ICN 2014

The Thirteenth International Conference on Networks

ISBN: 978-1-61208-318-6

February 23 - 27, 2014

Nice, France

ICN 2014 Editors

Tibor Gyires, Illinois State University, USA
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
György Kálmán, ABB Corporate Research, Norway

ICN 2014

Foreword

The Thirteenth International Conference on Networks (ICN 2014), held between February 23rd-27th, 2014 in Nice, France, continued a series of events focusing on the advances in the field of networks.

ICN 2014 welcomed technical papers presenting research and practical results, position papers addressing the pros and cons of specific proposals, such as those being discussed in the standard fora or in industry consortia, survey papers addressing the key problems and solutions, short papers on work in progress, and panel proposals.

We take here the opportunity to warmly thank all the members of the ICN 2014 Technical Program Committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to ICN 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the ICN 2014 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that ICN 2014 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of networks.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed the charm of Nice, France.

ICN Chairs:

Pascal Lorenz, University of Haute Alsace, France
Tibor Gyires, Illinois State University, USA
Eva Hladká, Masaryk University - Brno / CESNET, Czech Republic
Iwona Pozniak-Koszalka, Wroclaw University of Technology, Poland

ICN 2014

Committee

ICN General Chair

Pascal Lorenz, University of Haute Alsace, France

ICN Advisory Chairs

Tibor Gyires, Illinois State University, USA
Eva Hladká, Masaryk University - Brno / CESNET, Czech Republic
Iwona Pozniak-Koszalka, Wroclaw University of Technology, Poland

ICN 2014 Technical Program Committee

Alireza Abdollahpouri, University of Kurdistan - Sanandaj, Iran
Colin Allison, University of St Andrews, UK
Natalia Amelina, St. Petersburg State University, Russia | Norwegian University of Science and Technology, Norway
Pascal Anelli, University of Reunion, France
Jalel Ben-Othman, Université de Versailles, France
Max Agueh, LACSC - ECE Paris, France
Kari Aho, University of Jyväskylä, Finland
Pascal Anelli, Université de la Réunion, France
Cristian Anghel, Politehnica University of Bucharest, Romania
Jocelyn Aubert, Public Research Centre Henri Tudor, Luxembourg
Harald Baier, Hochschule Darmstadt, Germany
Alvaro Barradas, University of Algarve, Portugal
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Zdenek Becvar, Czech Technical University in Prague, Czech Republic
Djamel Benferhat, University of South Brittany, France
Ilham Benyahia, Université du Québec en Outaouais - Gatineau, Canada
Robert Bestak, Czech Technical University in Prague, Czech Republic
Jun Bi, Tsinghua University, China
Bruno Bogaz Zarpelão, State University of Londrina (UEL), Brazil
Fernando Boronat Seguí, Universidad Politécnica de Valencia, Spain
Agnieszka Brachman, Silesian University of Technology - Gliwice, Poland
Arslan Brömme, Vattenfall Europe AG | Corporate Security | Security Centre Germany - Berlin, Germany
Matthias R. Brust, Technological Institute of Aeronautics, Brazil
Bin Cao, Harbin Institute of Technology Shenzhen Graduate School, China
Jorge Luis Castro e Silva, UECE - Universidade Estadual do Ceará, Brazil
Joaquim Celestino Júnior, Universidade Estadual do Ceará (UECE), Brazil
Eduardo Cerqueira, Federal University of Para, Brazil
Marc Cheboldaeff, T-Systems International GmbH, Germany

Sun-il Kim, University of Alabama in Huntsville, USA
Wojciech Kmiecik Wroclaw University of Technology, Poland
Hideo Kobayashi, Mie University, Japan
Christian Köbel, Technische Hochschule Mittelhessen - Raum, Germany
André Kokkeler, Centre for Telematics and Information Technology, The Netherlands
Leszek Koszalka, Wroclaw University of Technology, Poland
Tomas Koutny, University of West Bohemi-Pilsen, Czech Republic
Polychronis Koutsakis, Technical University of Crete, Greece
Evangelos Kranakis, Carleton University, Canada
Francine Krief, University of Bordeaux, France
Michał Kucharzak, Wroclaw University of Technology, Poland
Radek Kuchta, Brno University of Technology, Czech Republic
Hadi Larijani, Glasgow Caledonian University, UK
Angelos Lazaris, University of Southern California, USA
Steven S. W. Lee, National Chung Cheng University, Taiwan R.O.C.
Jun Li, Qinghua University, China
Yan Li, Conviva, Inc. - San Mateo, USA
Feng Lin, Tennessee Tech University, USA
Diogo Lobato Acatauassú Nunes, Federal University of Para - Belem, Brazil
Andreas Löffler, Friedrich-Alexander-University of Erlangen-Nuremberg, Germany
Pascal Lorenz, University of Haute Alsace, France
Richard Lorion Université de la Réunion, France
Pavel Mach, Czech Technical University in Prague, Czech Republic
Damien Magoni, University of Bordeaux, France
Ahmed Mahdy, Texas A&M University - Corpus Christi, USA
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Anna Manolova Fagertun, Technical University of Denmark, Denmark
Gustavo Marfia, University of Bologna, Italy
Rui Marinheiro, ISCTE - Lisbon University Institute, Portugal
Antonio Martín, Seville University, Spain
Boris M. Miller, Monash University/ Institute for Information Transmission Problems, Australia
Pascale Minet, INRIA - Rocquencourt, France
Mohamed Mohamed, Mines-Telecom SudParis, France
Mario Montagud Climent, Universidad Politécnica de Valencia, Spain
Katsuhiro Naito, Mie University - Tsu City, Japan
Go-Hasegawa, Osaka University, Japan
Constantin Paleologu, University Politehnica of Bucharest, Romania
Konstantinos Patsakis, University of Piraeus, Greece
João Paulo Pereira, Polytechnic Institute of Bragança, Portugal
Kun Peng, Institute for Infocomm Research, Singapore
Ionut Pirnog, "Politehnica" University of Bucharest, Romania
Marcial Porto Fernandez, Universidade Estadual do Ceara (UECE), Brazil
Iwona Pozniak-Koszalka, Wroclaw University of Technology, Poland
Jani Puttonen, Magister Solutions Ltd., Finland
Shankar Raman, Indian Institute of Technology - Madras, India
Victor Ramos, UAM-Iztapalapa, Mexico
Priyanka Rawat, INRIA Lille - Nord Europe, France
Shukor Razak, Universiti Teknologi Malaysia (UTM), Malaysia

Vladimir Zaborovski, Polytechnic University of Saint Petersburg, Russia
Pavel Zahradnik , Czech Technical University Prague, Czech Republic
Arkady Zaslavsky, CSIRO ICT Centre & Australian National University - Acton, Australia
Sherali Zeadally, University of Kentucky, USA
Bing Zhang, National Institute of Information and Communications Technology - Yokosuka, Japan
Tayeb Znati, University of Pittsburgh, USA
André Zúquete, IEETA - University of Aveiro, Portugal

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

The Choice of VoIP Codec for Mobile Devices

K. Radnosrati
Converging Networks Laboratory
VTT Technical Research Centre of Finland
Oulu, Finland FI-90570
e-mail: kamiar.radnosrati@vtt.fi

D. Moltchanov and Y. Koucheryavy
Department of Electronics and Communications Engineering
Tampere University of Technology
Tampere, Finland FI-33101
e-mail: {moltchan,yk}@cs.tut.fi

***Abstract*—As modern wireless access networks are moving towards packet based wireless access one may expect mobile cellular telephony to be eventually replaced by voice-over-IP (VoIP) applications. The choice of the codec in these applications is not straightforward as packet-based power-aware wireless communications bring new factors into the play. We study interdependencies between the bitrate, energy consumption, and the perceived quality provided by the voice codecs. We show that it is sufficient to equip a software with three codecs only. These are G.729.E, G.711.1 and G.723.1 codecs. Among those, G.723.E provides the best trade-off between the involved factors. When the system is overloaded and/or the power consumption is the most important metric (i.e., a mobile is running out of power) G.723.1 provides the best possible capacity and energy savings at the expense of significant quality degradation. Finally, when the system is underloaded while the amount of power spent for running the service is not important G.711.1 provides the best possible heard quality at exceptionally high power consumption.**

Keywords-VoIP, energy conservation, codec, perceived quality.

I. INTRODUCTION

With the current generation of cellular wireless access technologies offering a wideband packet-based access over the air interface it is expected that voice-over-IP (VoIP) applications will be responsible for most part of the voice traffic. Although the full transition has yet to be made due to slow uptake of IP multimedia subsystem (IMS), forcing operators to use intermediate solutions such as circuit-switching fallback (CSFB), it will happen sooner than later or the customers may start moving towards third-party VoIP applications. the topic of perceived quality evaluation of VoIP codecs has been the subject of the study in [1], [2], and [3] among others. However, the main aim of this investigation is to describe the effect of loss correlation etc.

However, in addition to finalizing the convergence towards a unified all-IP multi-service network, this transition brings additional challenges to software developers. One of the choices that we need to make when developing a VoIP software is the type of the codec to use. The choice of the voice codec is more complicated in wireless environment as in addition to perceived quality provided to the user, one needs to take into account additional additional factors, such as power consumption and bitrate. There are a number of reasons for that. First of all, the uptime of mobile devices depends on their battery power that is evidently not growing at the pace of communication technologies implying that the chosen codec must be as energy efficient as possible. Secondly, wireless technologies are more prone to occasional packet losses that may affect the perceived quality provided by codecs differently. Indeed, when a codec with high compression ratio is used the amount of bandwidth required from the network is minimized while the data flow becomes very sensitive to packet losses. Conversely, for low compression ratios the bitstream is less sensitive to packet losses while the amount of required bandwidth becomes significant. Recall that the compression ratio affects the amount of energy required for both compression and transmission. Further, although the bitrate of most codecs are fairly low compared to the available capacity of modern cellular technologies, minimizing it is still an important issue for network operators, especially, in densely population areas.

Finally, in those applications where the type of the codec is allowed to be changed on-the-fly we are interested which codec maximizes a certain characteristic that can be important for current operational regime of a mobile. For example, when the battery of a mobile is running out of power while the voice session is currently ”on” we are interested in maximizing the battery lifetime at the expense of slightly degraded quality. Indeed, for a given wireless access technology, a certain codec is characterized by a certain amount of power consumption required for transmission. At the same time, the amount of energy required for compression depends on the hardware configuration only. Using codecs with different compression ratio affects both components differently. We will see that for WLAN technology these components are comparable making the choice of the optimal codec less obvious. Considering the abovementioned interdependencies, choosing the appropriate codec minimizing energy consumption of a device and maximizing quality provided to the user is a complex task.

In this paper, we carry out an in-depth study of interdependencies between the perceived quality measured by the objective performance metrics, energy consumption spent for encoding and transmission, and bitrate of the codec in wireless environment. Both of these metrics are modulated by two factors that are conventionally assumed to be independent of each other. These are the type of the codec and loss behavior of a channel. Our major finding are as follows (i) for adaptive systems G.729.E, G.711.1 and G.723.1 are sufficient to cover all regimes of a mobile (ii) for conventional regime of a mobile

 ISBN: 978-1-61208-318-6

G.729.E provides the best trade-offs between perceived quality, total energy consumption and the required bitrate (iii) for high-quality service one needs to use G.711.1 at the expense of exceptionally high total power consumption, (iv) in energy saving regime G.723.1 provide some rather insignificant performance gains over G.729.E. Also, it is important to note that all the studied codecs, except for plain a/μ-law G.711, are characterized by similar response to the packet losses implying that the best codec after compression remains the best after any amount packet losses. Finally, for IEEE WLANs, under a certain choice of parameters, the amount of energy spent for encoding is comparable to transmission power implying that the choice of the optimal codec depends on a given technology. In particular, G.729.E is no longer the optimal codec when operating in IEEE WLAN environment.

The paper is organized as follows. In Section II, we introduce the QoE metric we use in this paper. In Section III, we numerically evaluate those trade-offs involved in our study. Discussion on the optimal choice of the codec is provided in Section IV. Conclusions are given in the last section.

II. PERCEIVED QUALITY METRIC

Quality of VoIP codecs is evaluated at the application layer using specific tests developed for assessing the perceived speech quality. To perform these tests a number of methods have been suggested in the past. We distinguish between subjective and objective tests. Those tests involving surveying humans are called subjective tests. Objective tests are based on deriving applications layer performance metrics based on network performance parameters. These tests try to provide the relationship between network performance and subjective QoE metric.

Subjective metrics assessing quality of voice communications are mostly based on the mean opinion score (MOS) scale. MOS provides numerical indication of the quality of the voice after compression and/or transmission. The value of MOS is a number ranging from 1 to 5 with 5 corresponding the the best possible quality. MOS is estimated by averaging the results of a set of subjective tests, where a number of humans grade the heard audio quality of test sentences.

The widely recognized objective metric for VoIP applications is defined in the so-called E-model standardized by ITU-T [4]. According to E-model the psychoacoustic speech quality is defined as a non-linear additive function of different impairments. The measure of the quality is called an R-factor which is given by

$$R = R_0 - I_s - I_d - I_e + A, \quad (1)$$

where R_0 represents noise and loudness in terms of the signal-to-noise ratio at 0dBr point, I_s accounts for impairments occurring simultaneously with speech, I_d represents impairments that are delayed with respect to speech, I_e is the effect of special equipment, A is the advantage factor. Simply put, I_d is the delay of a packet, encoding impairments are included in I_s, while the compression and network losses are in I_e. The advantage factor accounts for special environments, where a user may sacrifice the quality with respect to availability of the service. The value of R-factor varies in between 0 and 100.

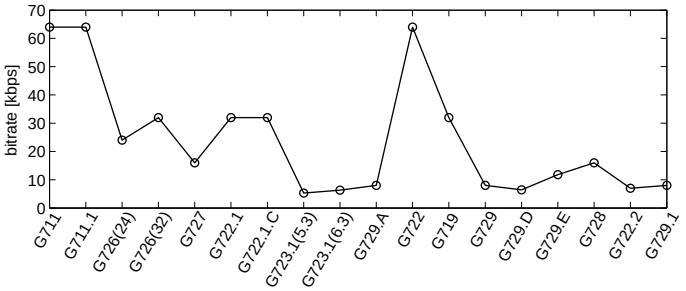


Figure 1: Raw bitrates of codecs.

Parameterizing the model we see that the advantage factor should nowadays be set to 0 as users get accustomed to wireless voice services. Following the work of Clark in [5], the highest possible value of $R_0 - I_s$ is set to 94 resulting in reduced expression $R = 94 - I_d - I_e$ setting the upper bound on the perceived quality. VoIP routes in the Internet are usually provisioned such that the end-to-end delay impairment factor, I_d, is less than the maximum tolerable delay ($150 - 200$ms, [7]). In this case the quality of speech transmission is dominated by I_e, i.e., $R = 94 - I_e$. The effect of I_e has been found using extensive subjective tests.

The performance of E-model was shown to correlate well with MOS grades under assumption of independent packet losses. In wireless networks, there are various mechanisms trying to remove the memory of the channel (e.g., interleaving). In wired Internet the major source of memory is droptail queuing. However, random earlier detection (RED) is gradually replacing droptail in the wired Internet making the packet loss process uncorrelated. Thus, the packet loss process in wireless-cum-wired configuration can be considered memoryless implying that there are no significant grouping of packet losses. It is important to note that this assumption can be relaxed whenever appropriate. For our work taking into account the effect of loss correlation would result in unnecessary increase of complexity and may hide the main message of the study.

III. INTERDEPENDENCIES

A. Rate requirements

Raw bitrates of codecs are shown in Fig. 1. It should be noted that some codecs have variable bit rates. For such codecs, we show only one of their data rates and the corresponding bandwidth. All the metrics we consider in what follows are calculated with respect to IP packet, i.e., energy consumption is expressed as mW per IP packet while rate is in IP packets per second.

The actual amount of data generated by a voice codec per sampling interval can be represented as $S = H + P$, bytes, where H is header size and P is the payload of voice packets. Further, denoting by the R the number of packets emitted per second and by the bandwidth, B, required for transmission is calculated as $B = S{*}R$ KBps. Thus, to estimate the bandwidth of a codec we need to know how much overhead, H, is added to the payload of a codec. When compressed real-time transport protocol (cRTP) is not used the IP/UDP/RTP headers amount up to 40 bytes due to the following components (i) IP header, 20 bytes, (ii) UDP header, 8 bytes, RTP, 12 bytes. In

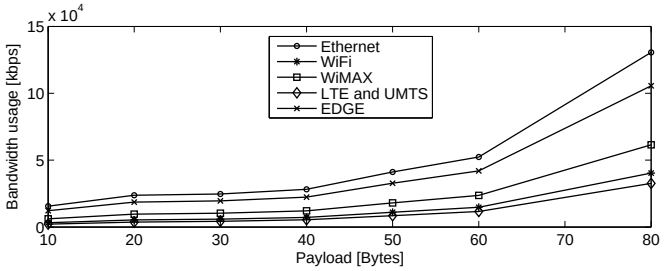


Figure 2: Bandwidth usage of audio codecs.

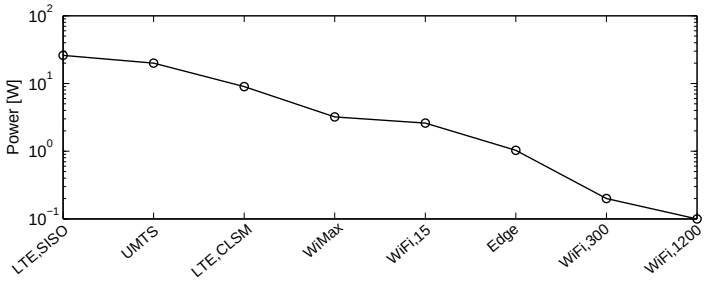


Figure 3: Energy consumption of transmission technologies.

those cases, when cRTP is used, the 40 bytes overhead reduced to just 2 bytes. In case of Ethernet, there will be additional 18 bytes that includes frame check sequence (FCS) and cyclic redundancy check (CRC) headers.

Fig. 2 shows the differences in bandwidth usage for different wireless access technologies and Ethernet. The payload size for these codecs changes between 10 and 80 bytes with step of 10 bytes. We see that the Ethernet imposes the highest bandwidth usage. Further, observe that the bandwidth requirements for all wireless access technologies presented here grow exponentially fast as the amount of payload increases. Out of all considered technologies EDGE requires the highest amount of bandwidth. The amount of bandwidth required by Wi-Fi and and Wi-Max technologies is comparable. LTE and UMTS requires the same amount of bandwidth and introduce the minimum overhead.

B. Energy consumption

1) Transmission energy: In wireless communications, the transmission power is different in different modes of operation. Conventionally, we distinguish between idle, sleep, transmission, and reception states. Here, we are interested in two of them, namely, transmission and reception states. Notices that it has been shown that that energy consumption in idle and receive states are almost the same [8]. The only difference between them is the amount of power spent by amplifying the received signal in receiving states. However, usually this energy is significantly smaller compared to that one required for transmission. Thus, we concentrate on the power consumption in transmission mode only.

Fig. 3 presents the power consumption measured in mW per packet for different wireless access technologies in logarithmic scale. As one may observe the difference between power consumption of wireless access technologies available today could be as high as two orders of magnitude proving the importance of choosing an appropriate radio interface and transmission technology one-the-fly. Another observation is the importance of choosing codecs. Indeed, different codecs produce their outputs in a wide range of data rates. Particularly, as we already observed the raw data rate ranges from 5.3Kbps for G.723.1 to 64kbps for G.722 or G.711.

2) Encoding energy: The energy spent for encoding varies with the type of the codec and its special features. Unfortunately, the actual energy depends on the type of a digital signal processor used for encoding. One way to provide

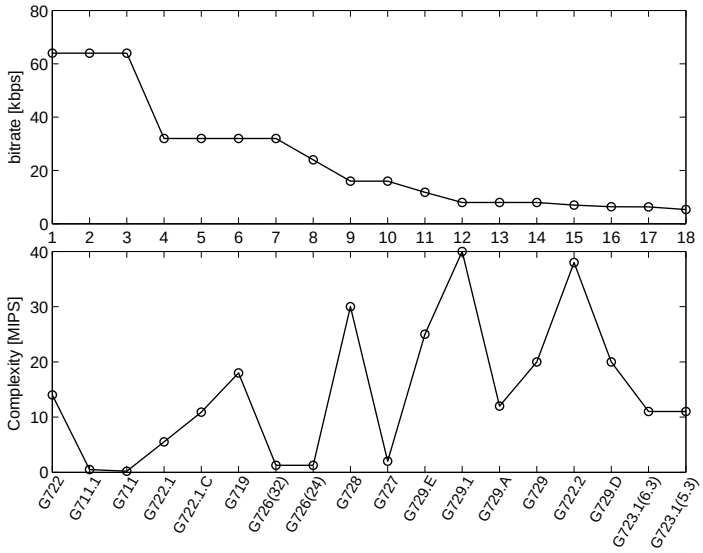


Figure 4: Complexity and bitrate for different audio codecs.

hardware independent estimates of the encoding complexity is to calculate the amount of operations required for encoding. Fig. 4 shows raw bitrate and complexity of voice codecs measured in millions operations per second (MIPS), where codecs are sorted in descending order of their bitrates. We see that the general trend is increase of the complexity in response to smaller bitrates. At one extreme there are G.711 and G.711.1 codecs having 64Kbps raw rates and requiring very small processing power (0.01mW for G.711 and 0.025mW for G.711.1 on C55x processors family). There are exceptionally complex codecs such as G.729.1 and G.722.2 requiring low raw rates. However, there are some exceptions, such as G.727, G.729 Annex A, and G.723.1 codecs characterized by rather low rates and moderate encoding power consumptions.

Investigations done in this paper show that the range of encoding energy for these codecs varies from below 1mW (G.711 with C55X) per packet to something around 12mW in some cases. Encoding power consumption for several processors is shown in Fig. 5. Note that for some codecs the amount of power required for encoding is comparable to the amount of power required for wireless access technologies. For example, the most complex codec G.729.1 running at C54x architecture requires a power of approximately 12mW, which is more than the amount of energy required for transmission

 ISBN: 978-1-61208-318-6

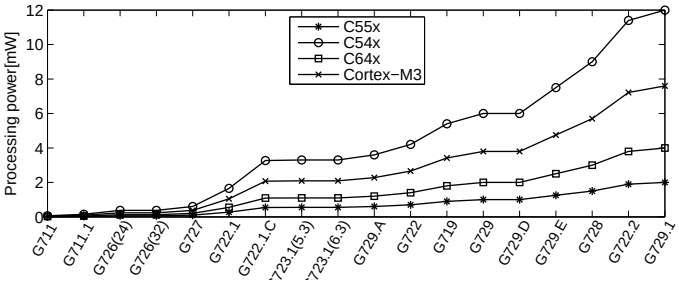


Figure 5: Encoding power consumption of voice codecs.

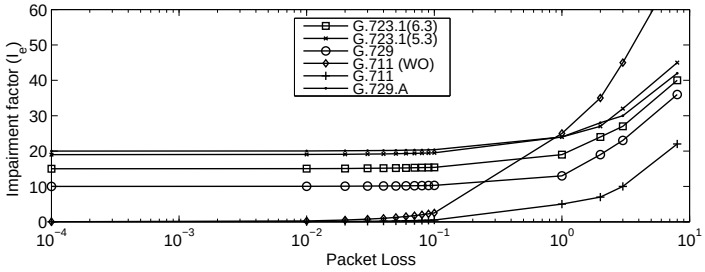


Figure 7: Values of I_e factor for different packet loss ratio.

over LTE in CLSM regime (approximately 9mW). On the other hand, low complexity codecs, such as G.711, G.711.1 requires exceptionally small amount of energy at all platforms. Note that the choice of the processor, plays an extremely important role and the difference in the encoding power could be as high as several mW. This difference becomes bigger as we go from low complexity codecs to the high complexity ones.

C. Perceived quality

Recall that the reduced E-model is given by $R = 94 - I_e$. The only unknown we have is I_e, which is the effective equipment impairment factor taking into account the effect of voice compression and network losses. I_e values for a number of considered codecs are summarized in Fig. 6 [6]. These values represent the perceived quality after compression and do not take into account the effect of packet losses introduced by the transmission medium. Recalling that R factor 94 is the maximum possible value achieved with G.711 while 70 is the minimum acceptable one the set of codecs available today provides the perceived quality across the whole range of acceptable quality.

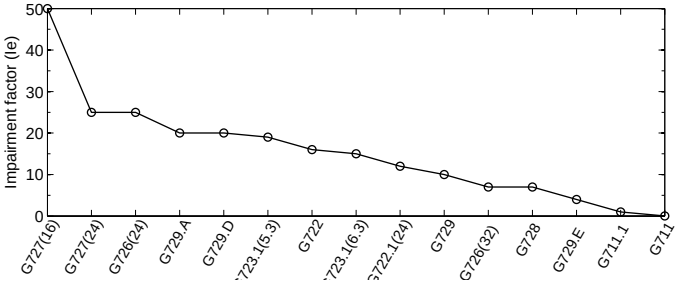


Figure 6: Performance degradation introduced by encoding.

Fig. 7 shows the values of impairment factors for a number of codecs for different values of the packet loss ratio (PLR, measured in percents). As one may observe almost all impairments factors I_e are linear functions of PLR. One of consequence of this behavior is that if the perceived quality is the only metric of interest then the choice of the best codec is independent of the packet loss ratio. In other words, the codec providing the best performance in absence of losses will remain the best one for any value of PLR. The only exception is G.711 codec without packet loss concealment feature whose performance severely degrade when PLR increases.

IV. Optimal choice of the codec

Consider how the amount of energy spent for compression is related to the power required for transmission. We would like to check whether there are special codecs minimizing the overall power consumption. The amount of power for different processor families required for encoding and transmission is shown in Fig. 8 and Fig. 9. As one may observe the transmission power for codecs is generally way larger than the energy required for compression. One special exception is Wi-Fi access technology operating with high TTI values, where these two sources of energy consumption are comparable for some codecs. Thus, in VoIP applications, for most wireless technologies transmission power dominates the overall power consumption.

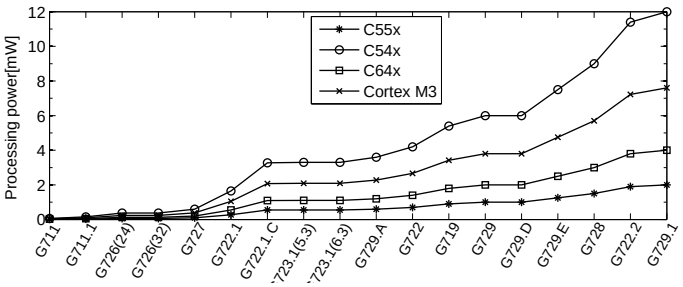


Figure 8: Encoding power consumption.

So far, we have seen that the trade-offs between perceived quality and energy required for running the service can be complicated. Putting all of these parameters together and viewing the results in one figure would be helpful. Fig. 10 shows total power consumptions of voice codecs per one second interval for C54x processor architecture (recall, that out of all considered platforms C54x requires the most power for encoding). Corresponding I_e values and R factors after compression are shown in Fig. 11. Assume for a moment that the bitrate of the codec is not a concern, i.e., the channel capacity is large enough to accommodate the one with the highest bitrate (G.711 or G.711.1). Even in this case, the choice of the codec optimizing both the total energy consumption and the perceived quality is still non-trivial. If one is targeting the best possible perceived quality G.711.1 is the obvious choice providing the maximum possible value of R-factor after compression. However, the amount of power spent for encoding and transmission is extremely high amounting to approximately 1250mW for UMTS and 1100mW for LTE CLSM. Energy requirements of G.711 is significantly smaller

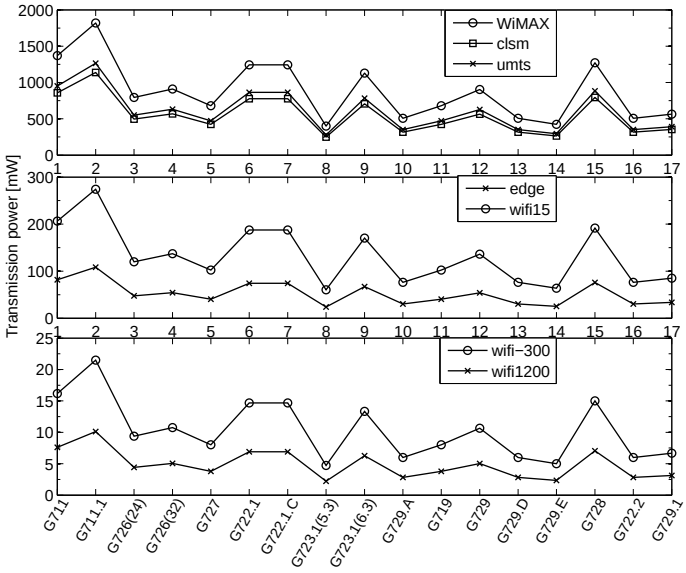


Figure 9: Transmission power consumption.

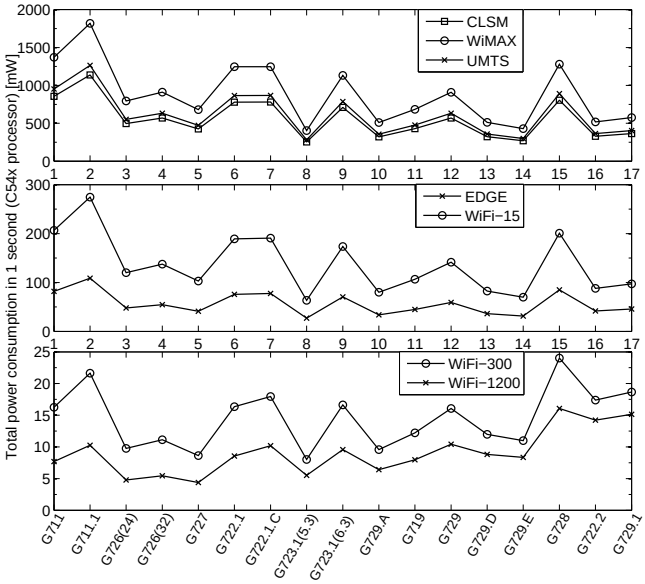


Figure 10: Total power consumption for C54x.

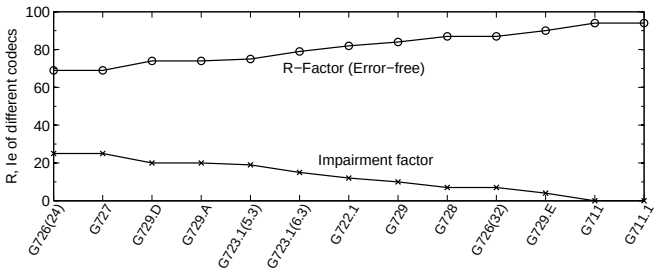


Figure 11: I_e and R-factor after compression.

(approximately 750mW for LTE CLSM and 900mW for UMTS) while the perceived quality is kept at the same level. Using G.711 instead of G.711.1 in loss-free environment would allows for save 350mW for both technologies. Assuming the average length of a conversation being equal to 300 seconds (5 minutes) it would result in approximately 100W energy savings. This implies additional 140 seconds of a VoIP call over LTE CLSM or approximately 116 seconds for UMTS VoIP call. However, these are not the best possible energy savings one may achieve. G.729.E codec providing R-factor of 90 while requires around 300mW of energy operating in LTE CLSM and UMTS networks. These significant additional energy savings compared to G.711 (approximately 450mW for LTE CLSM and 600mW for UMTS) comes at just slight decrease of quality (R-factor 90 compared to 94 for G.711). Further, taking into account the the data rate of G.729.E codec is just 12.5Kbps this codec is far superior compared to both G.711 and G.711.1. Energy consumption of of G.729.E codec is comparable to G.726 operating at 32Kbps data rate except for slightly worse R-factor (87 instead of 90). When energy is the most important metric (i.e., power of a mobile is running out) the best possible codec is G.723.1 operating at 5.3 Kbps data rate providing R value 75. This codec is especially useful for those wireless technologies characterized by small transmission energy requirements, i.e., IEEE 802.11 WLANs. However, as one may notice G.726 codec operating at 32Kbps provides significantly better quality (R-factor 87 compared to 79 for G.723.1). For energy saving regime the latter is advisable.

As we already highlighted, Wi-Fi operating with TTI 1200 is a special example of a wireless technology, where the energy consumption for compression is compared to that required for transmission. Observing Fig. 10 we see that the total power consumption for C54x processor family and Wi-Fi with TTI 1200 has a behavior different from other technologies. In fact, G.729.E codec is no longer the one providing one of the best trade-offs between the perceived quality and total power consumption for zero PLR. G.711 codec outperforms G.729.E in terms of total power consumption providing better perceived quality (7.5mW and R value 94 instead of 10mW and R value 90 for G.729.E). Moreover, G.726 codec operating at 24Kbps is even more (almost twice) energy efficient providing 5.5mW of total power consumption while delivering R value of 90. Note that similar conclusions are true for Cortex-M3 family while C64x family is closer to C55x performance with transmission energy dominating the total power consumption (not shown here). The choice of the optimal codec is more complex when the amount of energy for transmission and compression is comparable.

So far, we considered the trade-off between the amount of energy required for running the service and the perceived quality provided to the user. These factors are often enough when the system is well below its capacity limits. However, when it is about to overflow one also need to take into account the rate requirements of codecs as slight overload may lead to extreme quality degradation for all the users. When choosing the best possible codecs for these conditions one needs to take into account three factors simultaneously: rate requirements, perceived quality, and energy consumption. Taking another look at Fig. 10 and recalling data from Fig. 1 we see that G.729.E is still the best codec optimizing

 ISBN: 978-1-61208-318-6

these three parameters simultaneously (12.5 Kbps, R value 90, 400mW LTE CLSM and UMTS). G.728 operating at 16 Kbps, is characterized by comparable R value (87) but requires significantly more energy (approximately, 800mW and 900mW for LTE CLSM and UMTS for C54x, respectively). Similarly, G.726 operating at 32Kbps has slightly higher power requirements and comparable R value but requires approximately three times more bandwidth. When the system is severely overloaded G.723.1 operating at 5.3 Kbps is the best possible choice from the rate requirements perspective (more than twice better than G.729.E). However, its perceived quality is close to unacceptable (R value 79).

Consider now what happens when the PLR increases. Fig. 12 shows the values of R-factor and corresponding values of MOS for six selected codecs (computed according to the closed-form expression provided in [9]) for a number of codecs for different values of the packet loss ratio. Surprisingly, the choice of the VoIP codec jointly optimizing the considered three factors (rate/energy/quality) is independent of the value of PLR as response of the considered codecs to PLR is qualitatively and quantitatively similar. One exception is the very special behavior of G.711 codec without PLC capabilities whose R value decreases exponentially fast as PLR increases. This codec should never be used in lossy environments such as wireless access. Also, as one may observe, there is an intersection between lines corresponding to G.729.A and G.723.1 (5.3 Kbps) codecs, i.e., up to PLR of approximately 2.5% G.723.1 performs slightly better than G.729.A, while for higher value of PLR G.729.A outperforms G.723.1. This implies that if ones originally uses G.723.1, as PLR increases one needs to change to G.729.A. However, the region where G.729.A outperforms G.723.1 is below MOS 3.5, which is widely accepted as the minimum acceptable quality. Thus, in most cases, the choice of the best codec for non-zero value of PLR (after compression) coincides with the chose made for any non-negligible PLR.

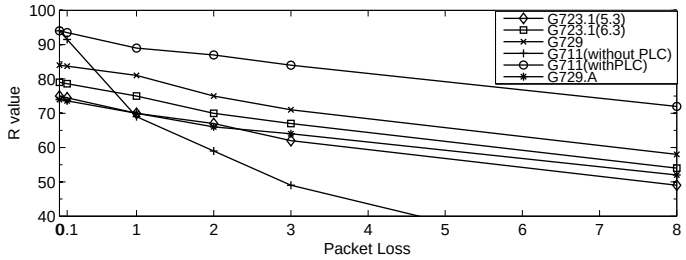


Figure 12: Values of R-factor for different packet loss ratios.

V. CONCLUSIONS

In this work, we examined trade-offs between quality of user experience, compression, and energy consumption for VoIP applications in wireless environment. As opposed to many studies exploring ways to minimize energy consumption of mobile devices we concentrated on the time period when a media application is up and running and studied the way how decrease the amount of energy required for running VoIP service while maintaining the best possible quality provided to the use. This study was motivated by availability of multiple codecs for voice and video information characterized by a wide diversity of compressed data rates and compression algorithms.

Summarizing, we note that the choice of the VoIP codec jointly optimizing the considered three factors (rate/energy/quality) is rather straightforward with only small deviations in the special cases. The reasons are (i) independence of the choice of the codec from packet losses (ii) domination of transmission energy requirements in total power consumption for most considered technologies (iii) one codec being significantly superior than others (G.729.E). From energy/quality/rate joint optimization point of view there are a number of obsolete codecs that are always worse compared to other. These are G.726 (24 Kbps and 32Kbps), G.727, G.729.D, G.729.A, G.723.1 (6.3 Kbps), G.722.1, G.729, G.728, G.711 (without PLC). The only codecs that need to be implemented to optimize the the abovementioned three parameters for any operational regime of a mobile are: G.729.E, G.711.1 (PLC mode), G.723.1 (5.3Kbps). The first codec, G.723.E, provides best possible performance for conventional regime of a wireless ensuring the best possible trade-off between the rate requirements (12.5 Kbps), power consumption (350mW LTE CLSM, UMTS), and the perceived quality (R value 90). When the system is severely overloaded and/or the power consumption is the most important metrics (i.e., a mobile is running out of power) G.723.1 operating at 5.3 Kbps provide the best possible capacity and energy savings (just 300mW for LTE CLSM and UMTS) at the expense of significant quality degradation (R value 75, which is close to the lowest possible quality level). This codec can only operate under zero PLR as any non-zero value of PLR immediately make the heard quality unacceptable. Finally, when the system is underloaded while the amount of power spent for communication is not important G.711.1 with PLC capabilities provides the best possible heard quality at exceptionally high power consumption.

REFERENCES

[1] M. Goudarzi, L. Sun, and E. Ifeachor, "Modelling Speech Quality for NB and WB SILK Codec for VoIP Applications, " Next Generation Mobile Applications, Services and Technologies (NGMAST), Cardiff, UK, 2011, pp. 42-47

[2] A. Marzuki, Y. Chai, H. Zen, L. Wee, K. Lias, and D. Mat, "Performance Analysis of VoIP over 802.11b/e using different codecs, " ISCIT 2010

[3] M. Aamir and S. Zaidi, "QoS analysis of VoIP traffic for different codecs and frame counts per packet in multimedia environment using OPNET," INMIC 2012

[4] G.107, "The E-model, a computational model for use in transmission planning", ITU-T Recommendation, 2003.

[5] A. Clark, "Modeling the Effects of Burst Packet Loss and Recency on Subjective Voice Quality," 2nd IPTel Workshop, 2001, pp.123-127.

[6] G.113, "Transmission impairments due to speech processing", ITU-T Recommendation, 2007.

[7] G.114, "One-way transmission time", ITU-T Recommendation, 1996.

[8] Y. Li, M. Reisslein and C. Chakrabarti, "Energy-Efficient Video Transmission Over a Wireless Link," IEEE Tran. Veh. Tech., vol.58, no.3, 2009, pp. 1229-1244.

[9] H. Assem, D. Malone, J. Dunne, and P. O'Sullivan, "Monitoring VoIP call quality using improved simplified E-model," Proc. ICNC, Jan. 28-31, San-Diego, USA, 2012, pp. 927-931.

 ISBN: 978-1-61208-318-6

Hybrid Cognitive Approach for Femtocell Interference Mitigation

Pavel Mach, Zdenek Becvar
Department of Telecommunication Engineering, Faculty of Electrical Engineering
Czech Technical University in Prague
Prague, Czech Republic
machp2@fel.cvut.cz, zdenek.becvar@fel.cvut.cz

***Abstract*—In this paper, we introduce new concept for femtocells with the purpose to minimize cross-tier interference to the macrocell users (MUEs). The cross-tier interference is mitigated by the power control algorithm minimizing the power level and, thus, ensuring that all currently active femto users (FUEs) can attach to the femto access points (FAPs). To guarantee quality of service to the FUEs even at heavy load, despite low transmitting power, the FAPs can opportunistically utilize also SU bands. To that end, we denote our scheme as a hybrid cognitive approach, which is distinguished by the fact that the FAPs can access bandwidth as a primary users (PU) and secondary users (SU) at the same time. In order to generate less overhead introduced by sensing for the purpose of determining available spectrum, we also propose an algorithm that adaptively changes a sensing period. The results show that our proposal is able to significantly reduce cross-tier interference nearly to the same level as achieved by algorithms focusing on maximization of the MUEs' performance. At the same time, our proposal ensures that the performance of the FUEs is kept at satisfactory level similarly as in the case of the algorithms focusing solely on the performance of the FUEs.**

Keywords-interference mitigation; femtocell; cognitive capabilities; sensing; power control.

I. INTRODUCTION

The Femto Access Points (FAPs) are small base stations deployed mostly indoor to cover locations with weak signal from a Macro Base Station (MBS) and to enhance performance of indoor users. The FAP can be classified depending on its access strategy into three types [1]: i) open access - the users have no restriction in accessing the FAPs, ii) closed access - only small group of users can connect to the FAPs, and iii) hybrid access a combination of both previous strategies. The FAPs can use either different frequency bands than the MBS (i.e., dedicated channel deployment) or share (fully/partly) the same bandwidth as the MBS (i.e., co-channel deployment) [2].

As pointed out by Hobby and Claussen [3], the main concern regarding the implementation of the FAPs is to guarantee that the macrocell users (MUEs) are not negatively affected by the FAPs, if closed access is considered and co-channel deployment is used. At the same time, the challenge is to achieve high performance of femtocell users (FUEs) to enable high data transmission within the coverage of the FAP. This is not a trivial problem, especially if a lot of FAPs are supposed to be deployed.

One feasible option to reduce interference is to use various power control approaches. Claussen et al. [4] propose three self-optimization schemes aimed to minimize interference to outdoor MUEs and to minimize number of generated handovers. Although the proposed power control can significantly mitigate the amount of handovers, the MUEs close to the FAP experience low SINR (Signal to Interference plus Noise Ratio). In addition, the self-optimization schemes do not always guarantee sufficient house coverage. Jo et al. [5] aim to minimize interference caused by the FAPs to passersby users, while providing sufficient indoor coverage. Yun and Cho [6] propose to adapt the FAPs power depending on the queue length at the FAP. If the queue length is low, the FAP decreases its transmitting power. In the opposite case, the FAPs power is increased in order to serve all generated data. In [7], we have designed Quality of Service guaranteed power control (QPC) algorithm. The transmitting power of the FAP is adapted not only according to current traffic load like in [6] but also according to the signal quality among the FUEs and the FAP in order to fully utilize data frame at physical layer. Still, schemes based on [6][7] are highly effective only for lower traffic loads. At heavy traffic loads, the power of the FAP has to be increased to satisfy all FUEs requirements and cross-tier interference rises as well.

Other eligible option for interference avoidance is to exploit cognitive radio and dynamic spectrum management [8]. Yu et al. [9] consider that the FAPs are able autonomously sense the radio frequencies used by the MBS and, thus, to schedule their transmission at unoccupied frequency spectrum. In addition, the optimal period for sensing of free radio resources is derived. Dynamic spectrum reuse in network with closed access FAPs is proposed by Demirdogen et al. in [10]. If the MUE is close to the FAP, the FAP equipped with cognitive radio decides whether to occupy the same frequencies or not (perfect sensing is assumed). Li and Sousa [11] view the FAP as a secondary system, which autonomously allocates orthogonal channels to avoid disturbing the MBS and its users. The objective to minimize interference among the FAPs is addressed by Li et al. in [12]. The FAPs cognitively recognize unoccupied frequencies and schedule their transmission accordingly. All above mentioned papers assume that the FAPs use only resources not currently utilized by the MBSs.

Rubaye et al. [13] exploit wider bandwidth by the FAPs than the MBS to support high demanding services in indoor environment. This work is further elaborated by Xie et al. in [14], where the idea is to use wider bandwidth by leasing available spectrum from other PSs. The authors consider

 ISBN: 978-1-61208-318-6

leasing available spectrum among the MBSs only. Then, the leased spectrum is dynamically assigned to the FAPs to support high demanding services in indoor environment. Both [13] and [14] are solely focused on performance of indoor users and the negative impact of the FAPs on the MUEs is not addressed.

In summary, the power control alone is not always able to mitigate cross-tier interference to the MUEs while satisfying QoS for the FUEs. At the same time, disadvantage of conventional cognitive approaches is that the amount of radio resources available to the secondary users (SUs) is highly dependent on activity of primary users (PUs). In this paper, the minimization of cross-tier interference is accomplished by power control, which sets transmission power of the FAPs only to such level that all active FUEs are able to connect to the FAP. In order to compensate for low transmitting power, we further introduce new concept, where the FAPs could access radio resources as a PU and SU simultaneously. Hence, if the PU bandwidth is not sufficient, the FAP can opportunistically use SU bands. The main advantage of this approach is that the FAPs are not fully dependent on other PU(s) and have always some radio resources at disposition. Since the FAPs are not able to transmit or receive any data when sensing is being performed [9], we try to maximize the profit obtained from the opportunistic usage of SU bands. To this end, we propose new algorithm minimizing the sensing time and sensing overhead, where the sensing period is adaptively changed depending on the loads of the FAP and other PUs.

The rest of the paper is structured as follows. The next section describes the system model together with problem formulation. Section III describes proposed power control algorithm and algorithm for dynamic change of sensing period. The simulation methodology and simulation results are addressed in Section IV and Section V, respectively. The discussion regarding simulation results are tackled in Section VI. The last section gives our conclusions and future work plans.

II. System Model and Problem Formulation

The system model considers three cellular operators (denoted as Operator A, Operator B, and Operator C) covering the same geographic area. The same frequency bandwidth is assigned to each operator. In order not to interfere with each other, allocated frequency bandwidths are not overlapping (i.e., each operator has dedicated its own bandwidth). We assume that the FAPs deployed in the area use primarily frequency bandwidth allocated to Operator A, i.e., the FAPs use this bandwidth as a PUs and this frequency band is referred to as a Primary Frequency (PF). Further, we assume that all FAPs have cognitive sensing capabilities and can utilize frequency bands of other two operators. Consequently, the FAPs use frequencies of Operator B and Operator C as SUs. In this paper, these bands are referred to as a Secondary Frequency (SF). Note that the SF band can be composed from more than one secondary system. In addition the FAPs could, in general, access not just frequency bands assigned to other operators but also other available free spectrum can be utilized for our purpose.

Our study is narrowed down, without loss of generality, to area covered by one MBS of every operator as indicated in Fig. 1. In this area, K femtocells are deployed transmitting with power $p_{t,1}, p_{t,2}, ..., p_{t,K}$. The FAPs are located close to the sidewalk and in the proximity of the MUEs. This scenario is very challenging in mitigating cross-tier interference, since the MUEs can experience low SINR. Note that SINR of the m-th MUEs (SINRm) is derived as

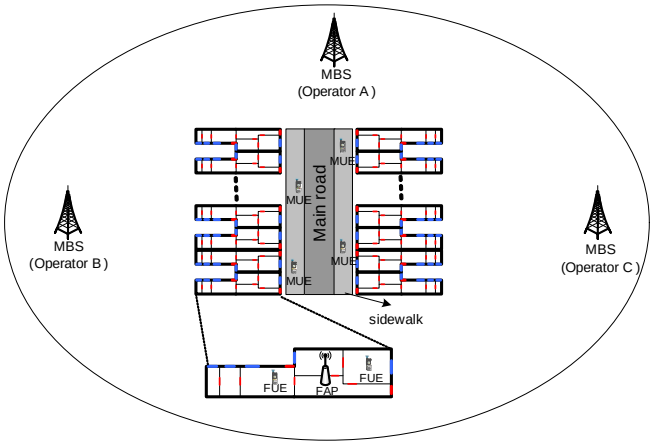


Figure 1. System model.

$$SINR_m = \frac{|h_{mk}|^2 \times P_t}{N + \sum_i^K |h_{mi}|^2 \times p_{t,i}}, \quad (1)$$

where $|h_{mk}|^2$ is a channel gain between the MUE and its serving MBS, $|h_{mi}|^2$ represents a channel gain between the MUE and the *i*-th FAP, P_t corresponds to the transmitting power of the MBS and N stands for the thermal noise. Similarly, SINR of the *f*-th FUE ($SINR_f$) is calculated as

$$SINR_f = \frac{|h_{ii}|^2 \times p_{t,i}}{N + \sum_{f \neq i} |h_{ft}|^2 \times p_{t,i} + |h_{fk}|^2 \times P_t}, \quad (2)$$

where $|h_{ii}|^2$ is a channel gain between the FUE and its serving FAP, $|h_{ft}|^2$ represents a channel model between the FUE and interfering FAPs and $|h_{fk}|^2$ corresponds to channel gain between the FUE and the MBS.

The maximum number of FUEs connected to one FAP is supposed to be up to four [15]. Since the access to the FAP is closed, only those FUEs belonging to FAP's Close Subscriber Group (CSG) can connect to it.

Both the MBS and the FAPs adopt OFDMA (Orthogonal Frequency Division Multiple Access) system based on LTE-A (Long Term Evolution-Advanced). The smallest amount of radio resources allocated for one user corresponds to a radio Resource Block (RB) occupying n_{SC} consecutive subcarriers in a frequency domain, and n_{SMB} OFDM (Orthogonal Frequency Division Multiplexing) symbols in a time domain. The RB is further decomposed to Resource Elements (REs) encompassing one subcarrier over one OFDM symbol. The amount of data transmitted in one RB for user j is derived as

$$n_{RB}^j = \Gamma \times \left(n_{SC} \times n_{SMB} - n_{OH}^{RE} \right), \quad (3)$$

where Γ is a transmission efficiency and n_{OH}^{RE} represents the number of REs in one RB dedicated for signaling overhead (e.g., reference signals or control information). The parameter Γ indicates the quantity of bits sent per RE and it depends on selected modulation and coding scheme (MCS) assigned according to the measured SINR. The amount of RBs per frame necessary for data transmission of m active FUEs is calculated as

$$n_{RB,r} = \sum_{j=0}^{m} ceil\left(\frac{\eta^j}{\eta_{RB}^j}\right), \quad (4)$$

where η^j is the amount of data sent in downlink (DL) for user *j*. The data is transmitted in a physical layer frame consisting of twenty RBs (m_{RB}) in time domain. In frequency domain, quantity of RBs per one frame (χ) depends on selected bandwidth (χ varies between 6 and 110). As a consequence, the amount of RBs available for a FAP during one frame within PF band is

$$n_{RB}^p = \chi \times m_{RB}. \quad (5)$$

Similarly, the amount of radio resources available in SF bands can be expressed as

$$n_{RB}^s = \left(\chi \times m_{RB} - n_{RB}^{B,u}\right) + \left(\chi \times m_{RB} - n_{RB}^{C,u}\right), \quad (6)$$

where $n_{RB}^{B,u}$ and $n_{RB}^{C,u}$ represent the quantity of RBs currently used by the MBSs of Operator B and Operator C. To decide, which radio resources at SF bands are not occupied, the FAPs are equipped with sensing functionality. The physical frames are sorted into sensing and data frames. The FAPs perform sensing solely during the sensing frames while the data frames are dedicated only for data transmission (including signaling). The overhead caused by the sensing of the *i*-th FAP is indirectly proportional to sensing period, T_s, and it is calculated as

$$\sigma^i = n_{RB}^p \times \frac{T_f}{T_s^i}, \quad (7)$$

where T_f represents length of one frame.

In general, the paper aims at addressing two problems. The first problem is to minimize FAPs transmitting power to avoid cross-tier interference to the MUEs. Thus, if we define P_{min} and P_{max} as the minimal and maximal transmitting power of the FAPs and $S_j = \{s_1^i, s_2^i, ..., s_g^i\}$ as a set of experienced SINR of q active FUEs connected to FAP *i*, our goal is to

$$min \sum_{i=1}^{K} p_{t,i} \quad (8)$$

$$s.t.\ s_i > SINR_{min} + \delta | \forall s_i \in S_i, \quad (9)$$

where $P_{min} \leq p_{t,i} \leq P_{max}$, $SINR_{min}$ represents the minimal value of SINR guaranteeing that the FUE is able to connect to the FAP and parameter δ stands for fading margin to have certain reserve to protect FUEs again fading effects.

The second problem is to minimize the sensing overhead and, thus, to maximize throughput for FUEs despite low transmitting power of the FAPs. Sensing overhead is a function of the current FUEs requirements ($n_{RB,r}$), the amount of radio resources in PF band (n_{RB}^p) and in SF bands (n_{RB}^s). Hence, the second objective is formulated as

$$min \sum_{i=1}^{K} n_{RB}^p \times \frac{T_f}{T_s^i} \quad (10)$$

$$s.t.\ T_{min} \leq T_s^i \leq T_{max}\ and\ T_s^i = f\left(n_{RB,r}, n_{RB}^p, n_{RB}^s\right). \quad (11)$$

III. Proposed scheme

The basic principle of the proposed scheme is depicted in Fig. 2. The FAP accesses the frequency band of Operator A as a PU. Note that the whole bandwidth of Operator A can be used by the FAP when compared, e.g., to [9] or [10], where only fraction of bandwidth, not occupied by the MBS, is available to the FAPs. To guarantee minimal cross-tier interference to MUEs of Operator A, the power of the FAP is set only to such level to maintain reliable connection between the FAP and its worst FUE (i.e., the FUE with the lowest SINR). Thus, the worst FUE has to utilize a more robust MCS (in Fig. 2, the FUE2 is the worst FUE and hence it uses 1/3 QPSK [16]). On the other hand, the radio channel between the FAP and the FUE1 is of a better quality. As a result, more efficient MCS is applied (in Fig. 2, 2/3 16QAM is shown as an example). The QoS of the FUEs would be degraded if only PF band is used by the FAP and if FUE's requirements exceed the number of available RBs at PF. An increase of power of the FAP can result in high cross-tier interference to the MUEs. Therefore, we rather suggest utilizing additional RBs of the Operator B and Operator C that are not currently used.

Our proposal is composed of two algorithms: power control algorithm and algorithm for dynamic adaptation of sensing period. These are described in detail in the next subsections.

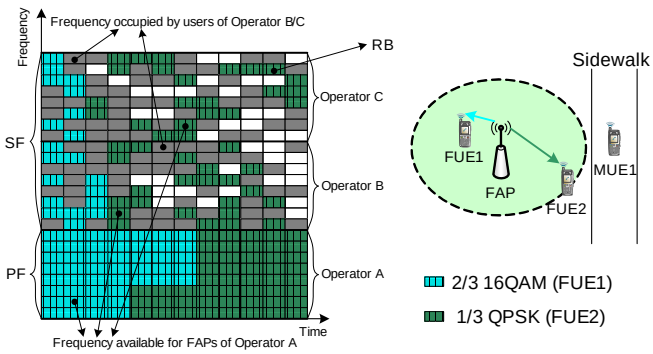


Figure 2. The example of proposed principle.

A. Power control algorithm

The flow chart of the power control algorithm is depicted in Fig. 3. First, the FAP discovers whether there is at least one active FUE. This is continuously determined after expiration of adaptation interval Δt. If no data are transmitted, the power of the FAP is set to its minimal value (P_{min}) to decrease a probability of interference to the close MUEs. In the opposite case, the transmitting power of the FAP is adjusted according to channel quality between the FAP and active FUEs. The transmitting power of the FAP is decreased by power adaptation step ΔP if

$$\forall s_j \in S_j |\ s_j > (SINR_{min} + \delta + \Delta P)\,. \tag{12}$$

The transmitting power after each iteration cannot exceed its minimal allowed value (P_{min}). Note that parameter ΔP serves as a kind of hysteresis to guarantee that the power is not decreased if the power was increased in the previous Δt. The transmitting power of the FAP is incremented by ΔP if

$$\exists s_j \in S_j |\ s_j < (SINR_{min} + \delta)\,. \tag{13}$$

Similarly as in previous case, the FAP can increase its transmitting power only to P_{max}.

The example of power adaptation according to the observed SINR is depicted in Fig. 4 where we assume, for the sake of clarity, that just one active FUE is attached to the FAP. At the beginning of the process, the transmitting power is set to P_{max} and subsequently is decreased until $s_j = SINR_{min} + \delta + \kappa$, where κ varies between 0 and ΔP (see Fig. 4). As long as the SINR is within allowed limits (i.e., $SINR_{min} + \delta < s^j < SINR_{min} + \delta + \Delta P$), the power remains the same. Note that slow fluctuation of SINR in Fig. 4 can be caused by slow movement of indoor users, not by different transmitting power of the FAP. The FUEs SINR can be temporarily not within allowed limits due to fading effects (in Fig. 4 indicated by variable η). If the SINR is suddenly increased, there is no harm for the FUE and transmitting power of the FAP is subsequently decreased. In the opposite case when the SINR is abruptly decreased, the outage can occur. This phenomenon is illustrated in Fig. 4 by sudden drop of SINR below $SINR_{min}$ during Δ_{out}. As a consequence, the transmission power of the FAP is increased step by step as long as all active FUEs experience sufficient SINR. The time when the FUE is in the outage needs to be minimized. This could be accomplished by proper setting of the power control parameters such as Δt, ΔP and δ. Due to space limitations and since the paper is rather focused on introduction of hybrid cognitive approach, the optimization setting of individual parameters is left for future research.

B. Algorithm for dynamic adaptation of sensing period

If radio resources at the PF band are not sufficient for the FUEs, the FAP can access also SF bands, if available. In order not to interfere with the MUEs of Operator B and Operator C, the FAP has to be aware of which RBs are currently utilized at SF bands. This is accomplished by sensing

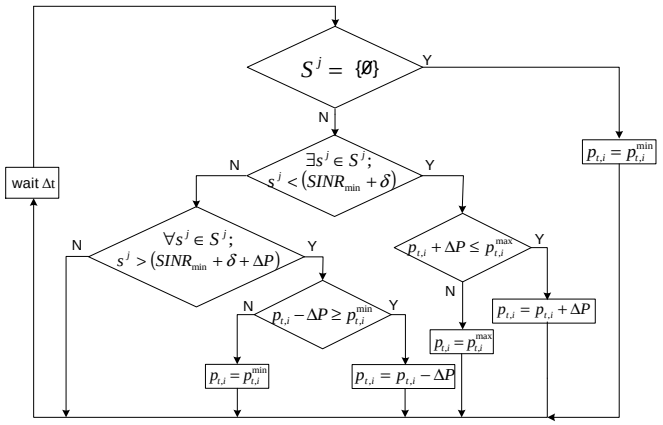


Figure 3. Flow chart of proposed power control algorithm.

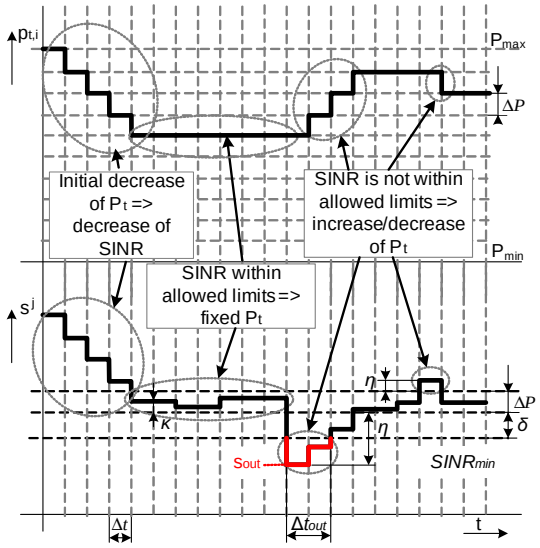


Figure 4. The example of power adaptation and its impact on observed SINR for j-th user.

the transmissions at SF to avoid interference. Contrary to [9], where fixed T_s is assumed we propose to set T_s dynamically depending on current traffic load of both the FAP and SF bands. The objective is to minimize the overhead generated by the sensing algorithm, i.e., to perform sensing only if it is profitable for the FAP.

The proposed algorithm for dynamic change of the T_s is depicted in Fig. 5. The sensing procedure is initiated after the power control is performed and once the required amount of RBs for DL transmission is derived. After the sensing is complete, it is evaluated if the amount of radio resources available for the FAP at the PF band is sufficient with respect to users requirements (i.e., if $n_{RB}^{p} > n_{RB,r}$). If this is the case, the T_s is set to its maximal value to minimize the sensing overhead. In this situation, the sensing is redundant, since the FAP uses only PF band. Nevertheless, it is still profitable to perform sensing occasionally to have overview regarding utilization of radio resources in SF bands. In this case, the algorithm sets Secondary Spectrum Usage Indicator (SSUI) to ”0”. The SSUI distinguishes whether the FAP allocates data only at PF or at both PF and SF.

 ISBN: 978-1-61208-318-6

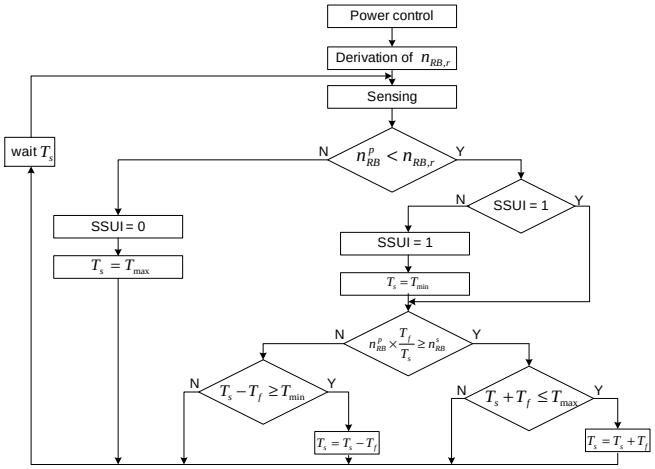


Figure 5. Flow chart of proposed sensing algorithm.

As soon as the amount of RBs available for the FAP at PF band is not sufficient (i.e., if $n^p_{RB} < n_{RB,r}$), the SSUI is switched to 1. After that, Ts is decreased to Tmin. This ensures that the FAP has up to date knowledge on utilization of the SF. Hence, the possibility of the FAP's interference to other PSs is minimized. On the contrary, the amount of generated overhead is increased. As long as the FAP requires additional radio resources to transmit all data, the length of Ts is dynamically changed depending on utilization of SF band by its PUs. Note that the sensing decreases the performance of the FAP if

$$n^p_{RB} \times \frac{T_f}{T_s} > n^s_{RB}, \quad (14)$$

since the sensing overhead is higher than available RBs at SF bands.

If the SF bands are currently overloaded and most of their radio resources are used, the proposed algorithm increases the length of sensing and, thus, decreases sensing overhead. The T_s is continuously increased until it is equal to T_{max} or until the load in SF band is decreased sufficiently. Contrary, T_s can be shortened when the amount of RBs at SF band is sufficient and, hence, more up to date knowledge of SF utilization minimize the probability of interference to other PSs.

The important aspect in sensing procedure is to estimate if RBs are utilized by other PSs or not. In this paper, we assume perfect sensing when the estimation whether RB is occupied or not is without errors similarly as in [10]. The reason is that the main purpose of this paper is not to propose new sensing techniques but we just exploit the ability of the FAP to perform the sensing.

IV. Simulation Setup

To evaluate the performance of the proposed scheme, simulations in MATLAB have been performed. Although the proposal is applicable to any OFDMA-based system, we use simulator based on FDD (Frequency Division Duplex) LTE-A (Release 10) with parameters set-up aligned with Small cell forum as presented in Table I. The movement of MUEs is restricted by sidewalks boundary (see Fig. 1). The MUEs are moving along straight trajectories from south to north with a speed of 1 m/s. Their distance from the house boundary is randomly generated with equal distribution between 1 and 3 meters. The intensity of MUEs arrival to the system follows Poisson distribution and it corresponds approximately to 140 passing users per hour. The movement of the FUEs within the house is based on [4]. At the beginning of the simulation, a start position for all FUEs is randomly selected at some waypoint (four FUEs are considered within each house). After that, the FUEs are moving along predefined trajectories between waypoints and points of decision as depicted in Fig. 7. The time spent by a FUE at the waypoint is described by normal distribution and differs for each room (parameters μ and σ of the distribution are also derived from [4]).

In the simulation, seven positions of the FAPs are selected in a distance varying between 1m to 7m from the house boundaries (see Fig. 6).

The path loss in indoor environment is calculated according to ITU-RP.1238 model. For evaluation of path loss in outdoor environment, COST 231 empirical model is used. Both selected path loss models are assumed, since these are widely used in the evaluation of femtocell concepts [17].

The amount of RBs available in SF band is indirectly proportional to traffic loads experienced by the MBSs of Operator B and Operator C. In the simulation, the traffic load of both MBSs varies between 50% and 100% with mean traffic load set approximately to 65%. This corresponds to the scenario when system is at heavy load state. The variation of the traffic load of the MBSs depends on activity/inactivity of MUEs of Operator B and Operator C. The MUEs change their status from active to inactive and vice versa by means of simple two state Markov model with the probability of 20%

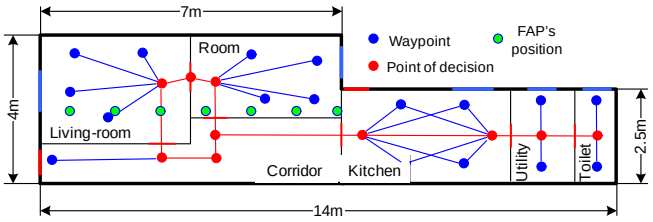


Figure 6. Indoor mobility model.

TABLE I. Parameter's Setting

Parameter	Value
Carrier frequency f [GHz]	2.0
MBS/FAP channel bandwidth BW (both primary and secondary bands) [MHz]	10/10
The number of RB in frequency domain χ [-]	50
Frame duration T_f [ms]	10
Max./min. FAP transmit power P_{max} / P_{min} [dBm]	21/-20
MBS transmit power [dBm]	43
Noise [W]	$BW \times 4 \times pW/GHz$
Number of FAPs/houses [-]	50/100
Loss of internal wall, external wall, window [dB]	5, 10, 3
δ [dB]	4
ΔP [dB]	2
Δt [ms]	10
T_{min},T_{max} [s]	0.2, 10
SINRmin [dB]	- 2 [16]
Physical layer overhead [%]	25

that the status of activity is changed. In addition, the outdoor MUEs are supposed to use more voice than data and, thus, voice is applied in 60% while FTP model is applied in 40%.

In the simulations, we have considered several performance metrics. The first one reflects the performance of FUEs, which is measured by served traffic in DL. In this case, model with mean generated traffic of 8.8 Mb/s is implemented in the simulations. The model is a combination of VoIP and FTP models according to [18]. The served traffic load (TL_s) in the performed simulation can be characterized as a difference between the generated traffic load (TL_g) and lost traffic load (TL_l) due to insufficient available radio resources formulated as

$$TL_s = TL_g - TL_l. \quad (15)$$

The second performance metric expresses the performance of the MUEs and it is measured by the transmission efficiency Γ. As already mentioned in Section II, Γ represents the amount of bits that could be sent through one RE. The Γ is derived from SINR, which sets the suitable MCS for data transmission. In our simulation, Γ is derived from SINR of the MUEs according to [16]. Note that the highest MCS (64QAM, coding 4/5) enables to transmit 4.8 bits per one RE [b/RE].

The third performance metric in our simulation is the sensing overhead (see (7)). The minimal value of T_s is set to 0.2 s, i.e., 20 frames. This value is selected in accordance with [9]. On the other hand, the maximal value of T_s is set to 10 s when sensing overhead is negligible, that is, 0.1%.

V. Results

The performance of our approach (in simulation labeled as Hybrid Cognitive Approach, HCA) is compared to the QPC scheme based on [7] and Cognitive Femtocell (CF) approach based on [9]-[12]. The QPC represents the case when the performance of the FUEs is maximized (in terms of served traffic) disregarding impact on the MUEs. On the other hand, the CF corresponds to a scenario when the FAPs use only radio resources not currently occupied by the MBS. Thus, the CF offers the highest performance to the MUEs even if the QoS of the FUEs could be worsened due to insufficient amount of radio resources at the side of the FAP. Note that in case of the CF, the FAPs transmission power is set according to simple auto-configuration scheme based on [4].

Fig. 7 illustrates the performance of outdoor MUEs moving along the sidewalk. The best results are achieved by the CF scheme, where performance is not negatively influenced by the FAPs as the FAPs use different radio resources. Consequently, the transmission efficiency is equal to 2.29 b/RE disregarding the FAPs location or generated load. Note that this value of Γ would be the same for the MUEs if no FAPs were introduced, since perfect sensing is considered and no interference is introduced to the MUEs. Further, it is obvious that the FAPs utilizing the QPC scheme cause the most significant interference to the MUEs as the MUEs experience DL transmission efficiency only in range between 1.02 to 2.07 b/RE. The situation is, in particular, unfavorable if the FAPs are located close to house boundaries. The performance of the

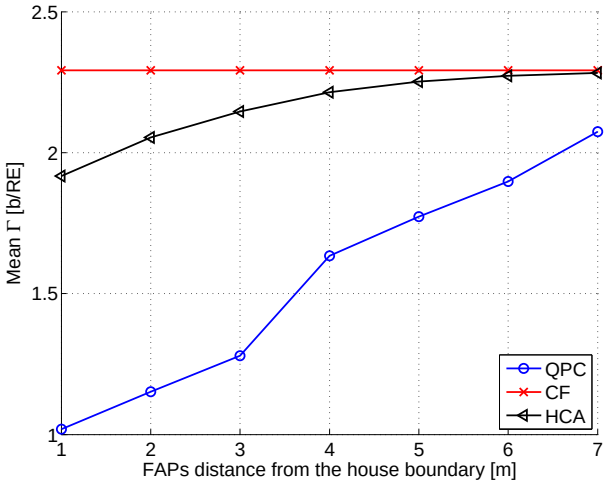


Figure 7. Mean transmission efficiency of MUEs.

HCA is only slightly lower than the performance of the CF in case that the FAP is located between 4 m and 7 m from house boundaries. But also for other cases the results are significantly better than the QPC.

Fig. 8 depicts the amount of served traffic for the FUEs. The highest amount of traffic is served in case of the QPC. This corresponds to the fact that the QPC tries, by all means, to satisfy the FUEs in terms of QoS. Consequently, approximately 97% of all generated data are successfully transmitted to the FUEs disregarding the FAPs position or generated load. On the contrary, the CF scheme is able to serve the lowest ratio of traffic varying only between 75% and 82.5%. The HCA scheme is able to ensure that up to 89% of generated traffic can be served indoor, which is substantially higher than in case of the CF.

Fig. 9 shows the influence of the amount of radio resources in SF band on traffic served by the FAPs. The simulations have been performed for all positions of the FAPs similarly as in Fig. 7 and Fig. 8 and subsequently the results were averaged out for individual loads. The performance of the QPC

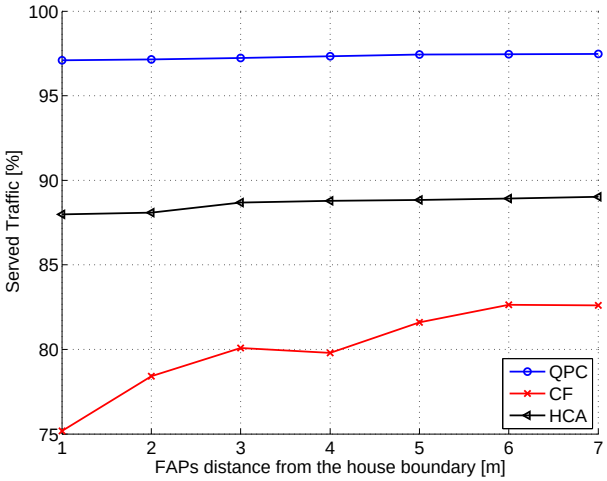


Figure 8. The amount of served traffic for FUEs.

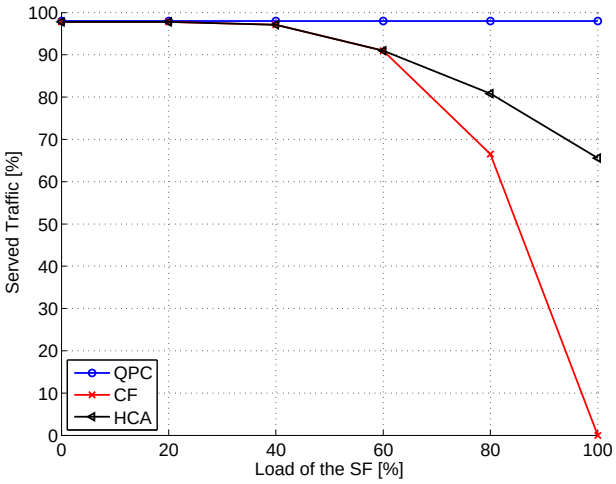


Figure 9. The amount of served traffic for FUEs depending on the load of the SF bands.

is the same as in Fig. 8, since it utilizes only PF band and no SF bands are taken into account. On the other hand, the CF performance is reliant only on the free radio resources in the SF band. As a consequence, the CF is able to serve less amount of traffic than the QPC. In the extreme case if the SF band is totally overloaded, no data can be transmitted by the CF scheme. On the other hand, the proposed HCA is not fully dependent on the utilization of SF bands by their users. Consequently, if no SF bands are available, still 65% can be served.

Fig. 10 presents sensing overhead generated by individual schemes. The highest sensing overhead is caused by the CF where fixed T_s is considered. In this case, 5% of FAP radio resources is required for the sensing. On the contrary, no sensing overhead is produced by the QPC since this scheme does not perform sensing at all. Regarding the HCA, the sensing overhead could be decreased approximately to 3.5%.

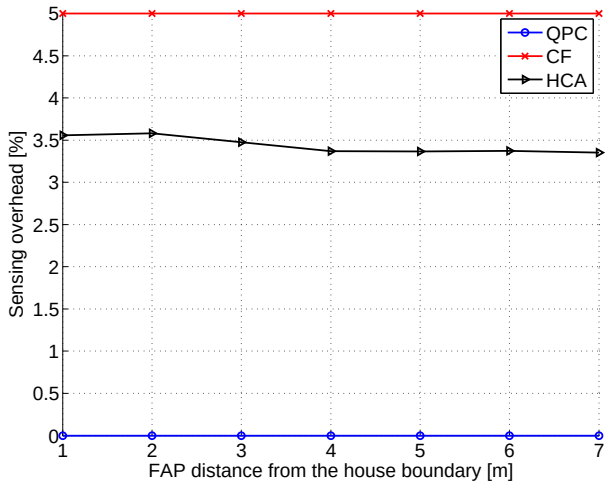


Figure 10. Comparison of overhead generated by sensing process.

VI. Discussion

The comparison of individual methods, when the results are averaged out over all FAPs location, is summarized in Table II. Regarding served traffic for the FUEs, the performance is the highest for the QPC, which serves 97.31% of traffic. Nevertheless, the main weakness of the QPC is its high interference to the MUEs as transmission efficiency is only 1.55 b/RE. Thus, applicability of the QPC is not feasible.

When the performance of MUEs is the main objective, the best results are achieved by the CF as the transmission efficiency is the highest (2.29 b/RE). On the other hand, the CF notably decreases performance of FUEs (only 80.05% is served), which is also not desirable, since the main purpose of the FAP is to enable high indoor data transmission. This problem can be further emphasized if the CF has not enough radio resources at SF. In this case, the CF fails and it is not able to serve FUEs at all (Fig. 9).

Based on above mentioned, the QPC is not suitable for the MUEs, whilst the CF is not sufficient from the FUEs point of view. The performed simulations indicate that the HCA is a convenient compromise between the QPC and the CF schemes. When compared to the CF, the transmission efficiency of MUEs is decreased only negligibly (2.16 b/RE). At the same time, the HCA ensures better performance for FUEs in comparison to the CF (88.61%). Also the sensing overhead is decreased in case of the HCA if compared to the CF.

As already suggested in Section 3A, the performance of the HCA can be further improved by optimization of several parameters such as Δt, ΔP and δ. The setting of individual parameters should be varying in dependence on the position of the FAPs. This way, the performance of the HCA in terms of transmission efficiency of the MUEs can be significantly improved to close the gap between the CF and the HCA. Similarly, the performance of the FUEs can be improved in terms of the FUEs served traffic to minimize a difference between the HCA and the QPC. The setting of optimal parameters is left to future research due to paper length limitation.

VII. Conclusion

paper has proposed a new hybrid cognitive approach where the FAPs can use both primary and secondary frequencies. Low interference to the MUEs is achieved by proposed power control algorithm. In addition, the minimization of sensing overhead is accomplished by algorithm that adaptively changes sensing period.

The results have been compared with two other competitive schemes. In overall, a disadvantage of the QPC is that the interference to the MUEs is significant while the CF degrades the performance of the FUEs, especially if the SF bands are heavily loaded. To that end, the proposed scheme offers a good

TABLE II. Comparison of individual methods

Scheme	MUEs Γ [b/RE]	FUEs served traffic [%]	Sensing overhead [%]
QPC	1.55	97.31	0.00
CF	2.29	80.05	5.00
HCA	2.16	88.61	3.44

 ISBN: 978-1-61208-318-6

trade-off between the MUEs and the FUEs performance. The results accomplished by the HCA can be further improved by optimization of power control parameters, such as adaptation interval, power adaptation step and fading margin. The optimization itself will be done in our future work.

ACKNOWLEDGMENT

This work has been supported by Grant No. 13-24932P funded by the Czech Science Foundation.

REFERENCES

[1] A. Golaup, M. Mustapha, and L. B. Patanapongpibul, "Femtocell Access Control Strategy in UMTS and LTE," IEEE Communication Magazine, vol. 47, 2009, pp. 117-123.

[2] V. Chandrasekhar and J. G. Andrews, "Spectrum Allocation in Shared Cellular Network," IEEE Transaction on Communication, vol. 57, 2009, pp. 30593068.

[3] J. D. Hobby and H. Claussen, "Deployment Options for Femtocells and Their Impact on Existing Macrocellular Networks," Bell Labs Technical Journal, vol. 13, 2009, pp. 145-160.

[4] H. Claussen, S. Pivit, and L. T. W. Ho, "Self-Optimization of Femtocell Coverage to Minimize the Increase in Core Network Mobility Signalling," Bell Labs Technical Journal, vol. 14, 2009, pp. 155-183.

[5] H. S. Jo, Ch. Mun, J. Moon, and J. G. Yook, "Self-optimized Coverage Coordination and Coverage Analysis in Femtocell Networks," IEEE Transaction on Wireless Communications, vol. 9, 2010, pp. 2977-2982.

[6] S. Y. Yun and D. H. Cho, "Traffic Density based Power Control Scheme for Femto AP," Proc. IEEE International Symposium on Personal Indoor and Mobile Radio Communication (PIMRC), 2010, pp. 1378-1383.

[7] P. Mach and Z. Becvar, "QoS-guaranteed Power Control Mechanism Based on the Frame Utilization for Femtocells," EURASIP Journal on Wireless Communications and Networking, vol. 2011, 2011, pp. 1-16.

[8] J. Marinho and E. Monteiro, "Cognitive Radio: Survey on Communications Protocols, Spectrum Issues, and Future Research Directions," Wireless Networks, vol. 18, 2012, pp. 147-164.

[9] L. S. Yu, T. Ch. Cheng, Ch. K. Cheng, and S. Ch. Wei, "Cognitive Radio Resource Management for QoS Guarantees in Autonomous Femtocell Networks," Proc. IEEE International Conference on Communications (ICC), 2010, pp. 1-6.

[10] I. Demirdogen, I. Guvenc, and H. Arslan, "Capacity of Closed-Access Femtocells Networks with Dynamic Spectrum Reuse," Proc. IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2010, pp. 1315-1320.

[11] Y. Li and E. S. Sousa, "Cognitive Femtocell: A Cost-Effective Approach Towards 4G Autonomous Infrastructure Network," Wireless Personal Communication, vol. 64, 2012, pp. 65-78.

[12] Y. Y. Li, M. Macucha, E. S. Sousa, T. Sato, and M. Nanri, "Cognitive Interference Management in 3G Femtocells," Proc. IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), 2009, pp. 1118-1122.

[13] S. A. Rubaye, A. A. Dulaimi, and J. Cosmas, "Cognitive Femtocells," IEEE Vehicular Technology Magazine, vol. 6, 2011, pp. 44-51.

[14] R. Xie, F. R. Yu, and H. Ji, "Energy-Efficient Spectrum Sharing and Power Allocation in Cognitive Radio Femtocell Networks," Proc. IEEE Annual International Conference on Computer Communications (INFOCOM), 2012, pp. 1665-1673.

[15] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, "Femtocell Networks: A Survey," IEEE Communication Magazine, vol. 46, 2008, pp. 59-67.

[16] Ch. Yu, W. Xiangming, L. Xinqi, and Z. Wei, "Research on the Modulation Coding Scheme in LTE TDD Wireless Network," Proc. International Conference on Industrial Mechanotrics and Automation (ICIMA), 2009, pp. 468-471.

[17] Small Cell Forum, "Interference Management in OFDMA Femtocells", 2011, accessed at http://www.smallcellforum.org (21 August 2013).

[18] ITU-R Tech. Rep. M.2135, "Guidelines for Evaluation of Radio Interface Technologies for IMT Advanced", 2008.

 ISBN: 978-1-61208-318-6

An Inter-domain Route Maintenance Scheme Based on Autonomous Clustering for Heterogeneous Mobile Ad Hoc Networks

Keisei Okano, Tomoyuki Ohta, and Yoshiaki Kakuda
Graduate School of Information Sciences, Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima 731-3194, Japan
Email: {okano@nsw.info, ohta@, kakuda@}hiroshima-cu.ac.jp

***Abstract*—Many types of mobile ad hoc networks such as vehicular ad hoc networks have been proposed for various application. In heterogeneous mobile ad hoc network environment that consists of many types of mobile ad hoc networks, each network uses a routing protocol suitable for the characteristics such as topology change frequency and data traffic. In such a network environment, nodes between different networks cannot communicate with each other because each network uses a different routing protocol. In this paper, we propose a new inter-domain routing protocol based on the autonomous clustering according to the network topology change in heterogeneous mobile ad hoc network environment and evaluate the effectiveness of the proposed scheme through simulation experiments.**

Keywords-Ad hoc network; Autonomous Clustering.

I. INTRODUCTION

There are many types of mobile ad hoc networks [1] such as vehicular ad hoc networks. Since the characteristics such as the topology change frequency and data traffic are different between networks, routing protocols that considered the characteristic of each network have been proposed for mobile ad hoc networks. Each network selects a suitable routing protocol from many routing protocols and uses it to enhance the network performance. As a result, in case that some mobile ad hoc networks exist in a region, it is possible that each network uses a different routing protocol. In such a heterogeneous mobile ad hoc network environment, there is no interoperability between networks and each network cannot communicate with each other so that each node cannot obtain much information and services even if much information and many services might exist in all networks. So far for inter-domain routing protocol in heterogeneous mobile ad hoc network environment, Cluster-based Inter-Domain Routing (CIDR)[2] and Inter-Domain Routing for MANETs (IDRM)[3] have been proposed. However, intra-routing protocol in each network is specified and it is difficult to select the routing protocol suitable for each network environment.

Therefore, this paper proposes an inter-domain routing protocol based on autonomous clustering to provide the communication between any two nodes in different networks for heterogeneous mobile ad hoc networks. In heterogeneous MANET environment, the network gateway nodes (shortly, NwGW nodes) are required to communicate between two different networks. In [4] and [5], we have proposed the two schemes to realize a new inter-domain routing protocol based on the autonomous clustering that we propose in this paper. In [4], we proposed the scheme to convert the control packet for providing the interoperability between two different network as ATR (Ad hoc Traversal Routing) and evaluate it in the environment where a specified number of nodes in the network is NwGW nodes. Next, in [5], we proposed the scheme to dynamically select the NwGW nodes between two different networks according to the network topology change. In this paper, we propose a route creation and maintenance scheme for the inter-domain routing protocol in heterogeneous MANET environment where the network topology change occurs frequently, and then evaluate it to show the effectiveness through simulation experiments.

The rest of the paper is organized as follows. In Section II, we describe requirements about our proposed scheme. In Section III, we introduce the proposed scheme itself. In Section IV, the experiments will be illustrated and the results will be discussed in the end.

II. REQUIREMENTS

In order to implement the proposed scheme, the mechanisms of ATR [4] and autonomous clustering [6], [7] in each node are required as a common platform. Each node has the routing protocol specified by the network on the common platform. In the heterogeneous mobile ad hoc network environment where some networks exist, each network is divided into multiple clusters and the nodes in the cluster is managed by the autonomous clustering. In the proposed inter-domain routing protocol, each cluster in the networks autonomously and dynamically selects one or more NwGW nodes from the nodes in the cluster, and then the source and the destination node in different networks can communicate with each other through NwGW nodes. In this time, the nodes which become NwGW nodes can forward any packets to nodes of the different network by using the mechanism of ATR so that the interoperability between different networks can be provided.

ATR [4] is the scheme to provide the interoperability between different networks in the heterogeneous mobile ad hoc network environment. Both any routing protocol and ATR work on each node. Each node converts from control packets which are used as the routing protocol in the network to control packets of ATR format, and forwards them to the neighboring node with ATR in the different network. The node with ATR that received the control packets of ATR format converts from

 ISBN: 978-1-61208-318-6

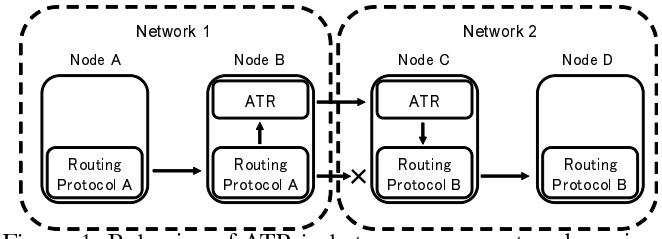


Figure 1: Behavior of ATR in heterogeneous network environment.

them to the control packets of a routing protocol used in the different network, and then forwards them in the different network. As a result, a node in a network can communicate with another node in a different network through nodes with ATR.

We explain ATR using an example as shown in Figure 1. Given that there are two networks, which are Network 1 and Network 2, and nodes A and B belong to Network 1 and nodes C and D belong to Network 2. When node A wants to communicate with node D, the route between nodes B and C cannot be created because the routing protocols are different. However, in this example, ATR works on nodes B and C so that nodes between Network 1 and Network 2 can communicate with each other through nodes B and C. Node B that receives a control packet of routing protocol A from node A converts from the control packet to a control packet of ATR, and then forwards it to node C. Node C that receives the control packet of ATR converts from the control packet to the corresponding control packet of routing protocol B, and then forwards it to node D. As a result, the route between nodes A and D can be created through nodes B and C.

A. Autonomous Clustering

Outline

Autonomous clustering [6], [7] is the scheme to divide the network into multiple clusters and manage nodes hierarchically. Each cluster consists of one cluster head, one or more gateways and cluster members. The cluster head manages nodes in each cluster and the cluster ID is assigned to the node ID of the cluster head. The gateway is neighbor to the nodes in the neighboring clusters. The packets are forwarded between clusters through gateways. In the autonomous clustering, the number of nodes in each cluster (that is, cluster size) is adjusted between the upper bound and lower bound given in advance.

Node State and State Transition

In mobile ad hoc network environment, nodes are always moving around the network so that the network topology is changed frequently. In the autonomous clustering, each node autonomously changes the state according to the situation of neighboring nodes to maintain the cluster. In the autonomous clustering, there are five states: CN, BN, BCN, NSN, and ON, and each node becomes one state of them and has a role in the cluster.

- CN (Control Node): This node works as a cluster head.
- BN (Border Node): This node works as a gateway.

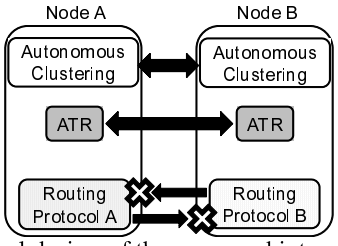


Figure 2: Protocol design of the proposed inter-domain routing.

- BCN (Border and Control Node): This node works as both a cluster head and gateway.
- NSN (Normal State Node): The node is a cluster member and does not work as a cluster head and gateway.
- ON (Orphan Node): This node does not belong to any clusters. In the initial state, a node becomes this state.

Cluster Configuration and Maintenance

The cluster head periodically broadcasts a control packet called MEP (MEmber Packet) within the cluster, and then cluster members that received the MEP sends MAP (Member Acknowledgment Packet) back to the cluster head. The cluster head can collect the information on cluster members and construct the cluster head-based tree in the cluster by these procedures.

The MEP includes the cluster ID and the node ID of the cluster head. The node that received MEPs stores the information and broadcasts it again. Each node receives MEPs from the neighboring node including the parent node and nodes of different clusters. Based on the received MEPs from the neighboring nodes, each node autonomously decides and changes its own state and cluster ID. For instance, if the cluster ID included in the received MEP is different from its own cluster ID, the node becomes gateway.

The nodes in the cluster that received the MEP sends a control packet MAP back to the cluster head as a reply. The MAP includes the cluster ID of the neighboring cluster, the node ID of gateways and the cluster ID of the neighboring cluster to which each gateway is neighbor, and the number of nodes in each neighboring cluster. The cluster head that receives MAPs from the cluster members manages the cluster by recognizing the number of nodes and the state of each node in the cluster as well as those in the neighboring clusters.

In order for the cluster head to maintain the number of nodes in the cluster between the upper bound and lower bound, the cluster head does the following procedures: it merges its own cluster with one of the neighboring clusters if the number of nodes is less than the lower bound and it divides the cluster into two clusters if the number of nodes is more than the upper bound.

III. An Inter-domain Routing based on Autonomous Clustering

A. Protocol Design

Figure 2 shows the protocol design of the proposed inter-domain routing protocol. In the proposed scheme, the autonomous clustering and ATR are required as a common

 ISBN: 978-1-61208-318-6

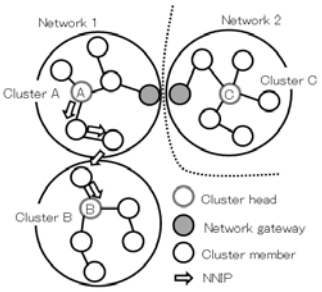


Figure 3: Connection status sharing among clusters.

platform in each node. The local routing protocol is a protocol to be used in each network.

In the proposed inter-domain routing, the communication between the source and the destination nodes in different networks can be provided with lower overhead and high data packet delivery ratio. In [5], we proposed the scheme to dynamically select the NwGW nodes between two different networks according to the network topology change.

B. Connection Status Sharing Mechanism

In heterogeneous MANET environment, since the route is created through NwGW nodes, the number of hops between the source and the destination nodes increases and the route break occurs more frequently. In the proposed inter-domain routing, when the route between the source and the destination nodes in different networks breaks, NwGW nodes try to repair the route. In order that NwGW nodes repair a route, each cluster heads share the connection status to the different network with neighboring clusters. The connection status to the different network consists of its own cluster ID, the network address of the neighboring network, and the number of hops to the cluster head of the neighboring cluster. Here, we explain how clusters share it with each other. In Section III-D, we describe the route repair mechanism based on the connection status.

After a cluster head sends a RNGP (Recommendation for Network Gateway node Packet), it periodically sends NNIP (Neighbor Network Information Packet) including its own connection status to the cluster heads of neighboring clusters. The cluster head that received NNIP from the neighboring clusters stores the connection status of the neighboring clusters.

We explain the the connection status sharing mechanism using Figure 3. As shown in Figure 3, given that there are clusters A and B in Network 1 and cluster C in Network 2. In this time, cluster head A recognizes the NwGW node (node E) that is neighbor to Network 2. Cluster head A adds (A, Network 2, 0) into NNIP, and then sends it to cluster head B. Cluster head B that received NNIP stores (A, Network 2, 4) into the neighboring cluster list. Here, the number of hops to the cluster head of the neighboring cluster contained in NNIP is changed from 0 to 4. This is because the hop count is incremented whenever the NNIP is forwarded at hop by hop based on the cluster head-based tree. Each cluster periodically exchanges the connection status with the neighboring clusters.

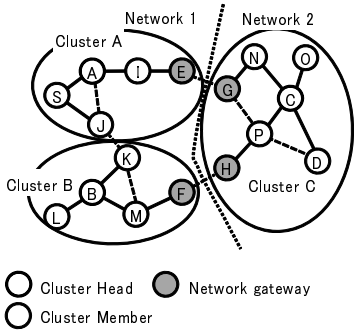


Figure 4: Network topology in heterogeneous MANETs.

C. Route Creation in Inter-Domain Routing

In heterogeneous MANETs, the route creation mechanisms in inter-domain routing are different according to the types of routing protocol in each network. In Figure 4, given that there are two networks in the field, and nodes S and D are a source node and a destination node. In this case, there are four cases, that is, (a) Networks 1 and 2 use reactive routing protocols, (b) Networks 1 and 2 use a reactive and a proactive routing protocol, (c) Networks 1 and 2 use a proactive and a reactive routing protocol, and (d) Networks 1 and 2 use proactive routing protocols.

(a) Networks 1 and 2 use reactive routing protocols

In this case, Network 1 to which the source node belongs and Network 2 to which the destination node belongs use a reactive routing protocol. The source node sends a route request message to the destination node by flooding. When NwGW nodes E and F in Network 1 receive the route request message, they convert it to the route request message of ATR and forward it to the neighboring NwGW nodes G and H. NwGW nodes G and H that received the route request message of ATR convert it to the route request message of the local routing protocol which is used in Network 2, and then forward it to nodes in Network 2. In case that the destination node D receives the route request message, it sends the route reply message toward the source node along the reverse route of the route request message.

(b) Networks 1 and 2 use a reactive and a proactive routing protocol

In this case, Network 1 to which the source node belongs uses a reactive routing protocol and Network 2 to which the destination node belongs uses a proactive routing protocol. The source node sends a route request message to the destination node by flooding. When NwGW nodes E and F in Network 1 receive the route request message, they convert it to the route request message of ATR and forward it to the neighboring NwGW nodes G and H. In case that NwGW nodes G and H that received the route request message of ATR have the route entry to the destination node, they send the route reply message to the source node.

(c) Networks 1 and 2 use a proactive and a reactive routing protocol

In this case, Network 1 to which the source node belongs uses a proactive routing protocol and Network 2 to which

 ISBN: 978-1-61208-318-6

the destination node belongs uses a reactive routing protocol. In case that the local routing protocol is a proactive routing protocol, NwGW nodes inform the neighboring network information of the local routing protocol. When the source node has the entry to NwGW nodes E or F that is neighbor to Network to which the destination node belongs, the source node forwards data packets to the NwGW node. The NwGW node that received data packets forwards them to NwGW node G or H in the neighboring network. The NwGW node that received data packets from NwGW node of the neighboring network sends a route request message by flooding in the network as a source node. When the destination node receives the route request message, it sends the route reply message to the NwGW node which is set as the designated source node. After the NwGW receives the route reply message, it forwards data packets to the destination node along the route.

(d) Networks 1 and 2 use proactive routing protocols

In this case, Network 1 to which the source node belongs and Network 2 to which the destination node belongs use a proactive routing protocol. A NwGW node can obtain the routing tables in the network from the local routing protocol, and then exchanges it with the NwGW node of the neighboring network. As a result, each node can add the route entry to the NwGW node which is neighbor to each network in the routing table. The source node forwards data packets based on the routing table.

D. Route Maintenance in Inter-Domain Routing

The route between the source node and the destination node is broken due to node movement. In case that the both nodes belong to an identical network, the route is repaired based on the local routing protocol which is installed in the network. However, in heterogeneous MANETs, it is impossible to repair the route based on one routing protocol because the route between the source node and the destination node is not created by one routing protocol. Therefore, the route repair procedures are different according to the location where the link was broken. There are three types of procedures for the route repair. The procedures are that (a) the route is broken in the networks that the destination node belongs to, (b) the route is broken in the networks the source node belongs to, and (c) the route is broken between two NwGW nodes in different networks. We explain the route repair procedure based on Figure 4. As shown in Figure 4, given that the source and destination nodes are nodes S and D, and the route is S, A, I, E, G, P, and D. In addition, since the procedure of a proactive routing is different from that of a reactive routing, we explain two types of procedures of reactive and proactive routings as (R) and (P).

(a) Route is broken in the networks that the destination node belongs to

In this case, given that the link between nodes P and D on the route is broken in Figure 4.

(R): Node P can detect the link break to the downstream nodes because of the notification of MAC protocol. After that, node P sends a route error message to the upstream node along the reverse route. NwGW node (node G) that received the route error message tries to recreate a new route to the destination node (node D) based on the local routing protocol in Network 2. In case that a route is recreated, node G restarts to forward data packets. Otherwise, node G sends the route error to the upstream nodes in the different network.

(P): After NwGW node G recognizes the route break based on the local routing protocol, it sends a route error message to the source node if it does not have the alternative route. Then, the source node tries to recreate a route to the destination node.

(b) Route is broken in the networks the source node belongs to

In this case, a node that detected the link break sends a route error message to the source node, and then the source node tries to recreate a new route to the destination node.

(c) Route is broken between two adjacent NwGW nodes in different networks

NwGW node (node E) that detected the route break forwards data packets to Cluster head (node A) and sets the timer. In case that NwGW node (node E) receives data packets until the timer is expired, it forwards data packets to the cluster head. The cluster head that received data packets from NwGW node (node E) forwards them to the neighboring cluster that is neighbor to the network including the destination node. Here, the neighboring cluster with the lowest number of hops to cluster head is selected from the neighboring cluster list. The cluster head (node B) that received data packets from the neighboring cluster forwards them to NwGW node (node F) in the same cluster, and NwGW node (node F) forwards data packet to NwGW node (node H) and tries to recreate the route to the destination.

(R): : If NwGW node (node H) has a route entry to the destination node, it forwards data packets to the destination node. Otherwise, NwGW node (node H) sends a route request by flooding only in the network to recreate a route. Data packets are forwarded to the destination node along the route in case that the route is created, while it sends a route error message to the source node in case that the route is not created. In this case, the source node that received the route error message recreates the route.

(P): : If NwGW node (node H) has a route entry to the destination node, it forwards data packets to the destination node. Otherwise, it sends a route error message to the neighboring NwGW node (node F).

After the timer on NwGW node (node E) is expired, it sends a route error message to the source node, and then the source node sends a route request message by flooding to recreate a route. In this case, since NwGW node (node F) has created the route entry to the destination node, it immediately sends the route reply message back to the source node, resulting in reducing the number of control packets.

 ISBN: 978-1-61208-318-6

TABLE I: Simulation environment.

Simulator	QualNet ver.5.0 [8]
Simulation time [s]	300
Number of nodes	200
Number of neighboring nodes	8, 10, 12
Transmission range [m]	250
Node moving speed [m/s]	10, 20
Number of transmitted data packets	1000
Data packet size [byte]	512
Interval of sending data packets [s]	0.25
Number of pairs of source and destination nodes	10
Node mobility model	Random Waypoint Model
Maximum cluster size	50
Minimum cluster size	10
Interval of sending MEP [s]	2
MAC protocol	IEEE802.11b

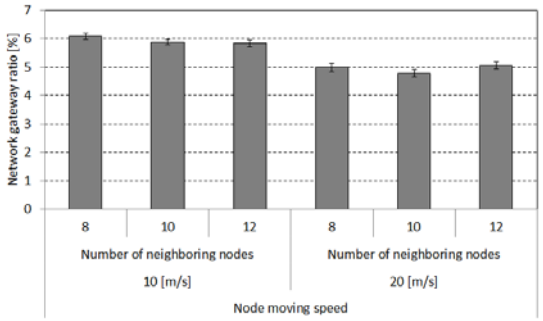


Figure 5: NwGW node ratio.

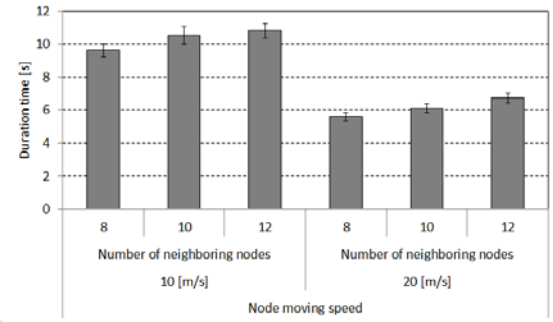


Figure 6: Duration time of two NwGW nodes.

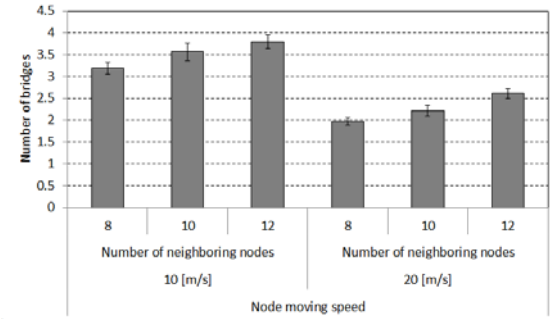


Figure 7: Number of bridges.

IV. Simulation Evaluation

A. Simulation Plan

Table I shows the simulation environment. In heterogeneous MANET environment, there are two networks, which are Network 1 and Network 2. Although both networks use AODV [9], in the simulation each routing protocol is handled as a distinct routing protocol and both networks cannot communicate with each other. In each pair of a source and a destination node (SD pair), the source and the destination node belong to Network 1 and Network 2, respectively. In addition, each cluster selects one NwGW nodes for each network by the dynamic network gateway selection scheme.

Evaluation criteria are the NwGW node ratio, the duration time of two NwGW nodes, and the number of bridges. The duration time of two NwGW nodes is the time when two NwGW nodes in different networks are continuously neighboring. The number of bridges is the number of pairs of two network NwGW nodes between networks. In addition, in order to show the effectiveness of the route maintenance, we show the results of the data packet delivery ratio and control overhead by comparing between w/ route repair and w/o route repair.

B. Simulation Results

NwGW node ratio

Figure 5 shows the NwGW node ratio. From Figure 5, it is confirmed that the NwGW ratio is almost the same regardless of the number of neighboring nodes.

Next, in case that the node moving speed is 20 m/s, the NwGW node ratio decreases in comparison with the node moving speed is 10 m/s. The NwGW nodes are selected from cluster members and notified to a selected cluster member based on MAPs which are sent by cluster members. However, there is the difference between the time when cluster members send MAPs to the cluster head and the time when the cluster head selects a NwGW node. Therefore, due to node movement, there is the possibility that the new selected NwGW move out from the clsuter. In this case, the NwGW node is not selected, and then another new NwGW node is selected next time. As a result, as the node moving speed becomes faster, the NwGW node ratio becomes lower.

Duration time of two NwGW nodes

Figure 6 shows the duration time of two NwGW nodes. In these experiments, we set at the smaller field size in order to increase the number of neighboring nodes. Therefore, two NwGW nodes in different networks are adjacent at the high possibility. In addition, as the node moving speed becomes faster, the relative speed of two NwGW nodes becomes faster and the duration time of two NwGW nodes becomes shorter.

Number of bridges

Figure 7 shows the number of bridges. As shown in Figure 7, the number of bridges increases as the number of neighboring nodes becomes more. However, in case that the node moving speed is 20 m/s, the number of bridges decreases because the number of NwGW nodes decreases as shown in Figure 5.

Data packet delivery ratio

Figure 8 shows the data packet delivery ratio. In all cases, the scheme w/ route repair has the higher data packet delivery ratio than the scheme w/o route repair. In case that the node moving speed is 20 m/s and the number of neighboring nodes

 ISBN: 978-1-61208-318-6

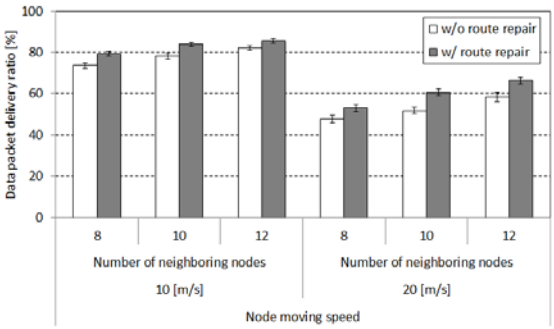


Figure 8: Data packet delivery ratio.

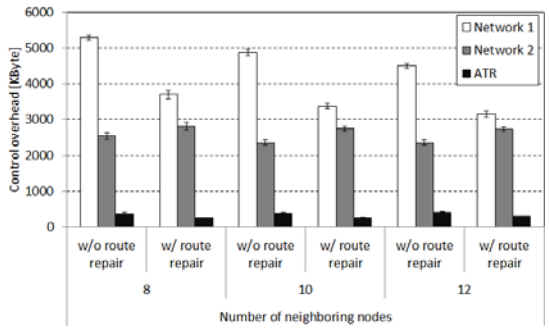


Figure 9: Total control overhead in case of node moving speed 20 m/s.

is 10, the scheme w/ route repair becomes 8.7% higher than the scheme w/o route repair. In this case, the route break occurs more frequently in comparison with the case of node moving speed 10 m/s. Therefore, as shown in Table II, the route repair is frequently invoked by not a source node but NwGW nodes.

On the other hand, in case that the node moving speed is 10 m/s and the number of neighboring nodes is 12, there is only 3.2% difference between the scheme w/ route repair and w/o route repair. Since the node density is high, the route repair is quickly invoked by the source node.

As a result, in case that the node moving speed is fast and the route breaks often occur, the proposed route maintenance scheme behaves efficiently and provides the high data packet delivery ratio.

Total control overhead

Figure 9 shows the total control overhead, which does not include the control overhead of the autonomous clustering. In cases of Network 1 and ATR, the control overhead of the scheme w/ route repair becomes 30% lower than that of the scheme w/o route repair. On the contrary, in case of Network 2, the control overhead of the scheme w/ route repair becomes 10% higher than that of the scheme w/o route repair. This is because in case of the scheme w/o route repair, when the route break occurs in Network 2 to which the destination node belongs, the source node sends a route request message by flooding in all network to recreate a route. On the contrary, in case of the scheme w/ route repair, when the route break occurs in Network 2, the NwGW node sends a route repair message by flooding only in Network 2 and the messages are not flooded in Network 1. However, if the NwGW cannot recreate a new route to the destination node in Network 2, the NwGW node

TABLE II: Number of route creations invoked by NwGW nodes

# of neighboring nodes	Moving speed [m/s]	# of route creation	Success	Failure
8	10	472	456	16
	20	588	563	25
10	10	444	437	7
	20	602	587	15
12	10	411	405	6
	20	592	582	10

sends a route error message to the source node, and then the source node tries to recreate a new route. In this case, a route request messages is flooded in all networks. Therefore, it is considered that the control overhead of the scheme w/ route repair increases in comparison with that of the scheme w/o route repair.

V. Conclusion

This paper has proposed an inter-domain routing protocol based on autonomous clustering for heterogeneous MANETs and evaluated it through simulation experiments. From simulation experiments, it is confirmed that the route repair mechanism works more effective especially in case that the network topology change occurs more frequently. In the future work, we are planning to repair a route in a shorter time and become higher data packet delivery ratio with lower overhead.

Acknowledgments

This work was supported in part by JSPS KAKENHI (Grant Number 24700073 and 24300028), the Hiroshima City University under Grant for Special Academic Research, and MIC SCOPE (No.131408006).

References

[1] C.-K. Toh, “Ad Hoc Mobile Wireless Networks Protocols And Systems, ” Prentice Hall Inc., 2002.

[2] B. Zhou, Z Cao, and M Gerla, ”Cluster-based inter-domain routing (CIDR) protocol for MANETs,” Proc. 6th Int’l Conf. on Wireless On-Demand Network Systems and Services (WONS), pp.19-26, Feb. 2009.

[3] C.-K. Chau, J.Crowcroft, K.-W.Lee, and S.H.Y. Wong, ”Inter-domain routing for mobile ad hoc networks,” Proc. 3rd ACM Int’l Workshop on Mobility in the evolving internet architecture (MobiArch’08), pp.61-66, Aug. 2008.

[4] S.Fujiwara, T.Ohta, and Y.Kakuda, ”An inter-domain routing for heterogeneous mobile ad hoc networks using packet conversion and address sharing,” Proc. 32nd IEEE Int’l Conf. on Distributed Computing Systems Workshops (ADSN 2012), pp.349-355, June 2012.

[5] K.Okano, T.Ohta, and Y.Kakuda, “A dynamic network gateway selection scheme based on autonomous clustering for heterogeneous mobile ad hoc network environment, ” Proc. IEEE Global Communications Conference Workshop (GC’12 Workshop), pp.513-517, Dec. 2012.

[6] T.Ohta, S.Inoue, and Y.Kakuda., ”An adaptive multihop clustering scheme for highly mobile ad hoc networks,” Proc. 6th IEEE Int’l Symp. on Autonomous Decentralized Systems (ISADS2003), pp.293-300, April 2003.

[7] T.Ohta, S.Inoue, Y.Kakuda, and K.Ishida, ”An adaptive multihop clustering scheme for ad hoc networks with high mobility,” IEICE Transactions on Fundamentals, vol.E86-A, no.7, pp.1689-1697, July 2003.

[8] “Qualnet network simulator by scalable network technologies, ” http://www.scalable-networks.com/, 2012.

[9] C. Perkins, E. Belding-Royer, and S.Das, “Ad Hoc On-Demand Distance Vector (AODV) Routing, ” IETF RFC3561, 2003.

 ISBN: 978-1-61208-318-6

A Mobile Agent-based Service Collection and Dissemination Scheme for Heterogeneous Mobile Ad Hoc Networks

Shuhei Ishizuka, Tomoyuki Ohta, and Yoshiaki Kakuda
Graduate School of Information Sciences, Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku, Hiroshima 731-3194, Japan
Email: {ishizuka@nsw.info, ohta@, kakuda@}hiroshima-cu.ac.jp

***Abstract*—A mobile ad hoc network (shortly, MANET) that consists of mobile nodes (shortly, nodes) is one of autonomous decentralized networks. Since the network topology changes frequently due to the node movement, it is difficult for each node to grasp the application (service) in MANETs. In order to solve this problem, the service information discovery scheme using mobile agents has been proposed for MANETs. In this scheme, a mobile agent collects and disseminates service information while moving autonomously from a node to another node in the network. However, in heterogeneous MANETs, mobile agents cannot migrate between different MANETs. Therefore, in this paper, we propose the mechanism to efficiently collect and disseminate service information based on mobile agent, and then show the effectiveness of the proposed scheme through simulation experiments.**

Keywords-Ad hoc network; Service information; Mobile agent.

I. INTRODUCTION

A Mobile Ad Hoc NETwork (MANET) [1] is a wireless distributed network that consists of only mobile nodes without the aid of the access points and fixed infrastructure. When a node wants to communicate with another node outside the transmission range, both nodes can be communicated with each other through intermediate nodes between the two nodes. Since a MANET is easily configured by only mobile nodes, many types of MANETs are used for a variety of application. The number of nodes in a MANET, the frequency of the topology change or the total volume of data traffic is different among multiple MANETs. Therefore, a routing protocol that is appropriate for each MANET is different and each MANET uses a different routing protocol because the characteristics of each MANET is different. However, in this case, even if multiple MANETs coexist nearby in an area, MANETs cannot communicate with each other because routing protocols are different. As a result, each node cannot obtain many information that could obtain from nodes in different MANETs. One idea to solve this problem is that all MANETs use the identical routing protocol. However, in this case, it is expected that the performance may degrade in each MANET. The other idea is to deploy the network gateway between MANETs to connect with each other like the Internet. Therefore, we have proposed the mechanism to select the network gateways to provide the interoperability between different MANETs in [7]. The selected nodes serve as the network gateway, but the network gateways change with time because nodes are always moving in MANETs. In this paper, we define the network that multiple MANETs coexist in an area and each MANET can communicate with another MANET as a heterogeneous MANET environment. In addition, there is no network administrator to manage all services, it is difficult for nodes to discover the services in the network because the network topology changes. Therefore, many service discovery schemes [2], [3] have been proposed for the mobile ad hoc network environment.

In this paper, we propose a mobile agent-based service collection and dissemination scheme and a new node architecture for the proposed scheme, and then show the effectiveness of the proposed scheme through simulation experiments. In this scheme, mobile agents collect and disseminate service information that each node holds while migrating from a node to another node, and we have shown that mobile agents can efficiently work in MANETs [5]. In the heterogeneous MANET environment, as the number of network gateways in the network increases, much more links to connect between MANETs are configured, but the overhead becomes much higher. Therefore, in [7], the number of network gateways becomes as low number as possible in comparison with the tonal number of nodes in the network. A service information in a MANET must be forwarded to another MANET through the network gateways between these MANETs. Therefore, mobile agents have to migrate from a node to another node while considering the location of the network gateway nodes that are dynamically changed in heterogeneous MANET environment.

The rest of the paper is organized as follows. Section II presents the node architecture in the proposed scheme. Section III shows the proposed scheme in more detail. In Section IV, we evaluate the proposed scheme and show the simulation results. Finally, we conclude this paper in Section V.

II. NODE ARCHITECTURE FOR HETEROGENEOUS MANETS

In order to implement the proposed node architecture, the mechanisms of autonomous clustering [4] and ATR(Ad hoc Traversal Routing) [6] in each node are required as a common platform. Each node has the routing protocol specified by the network on the common platform. In the heterogeneous mobile ad hoc network environment where some networks exist, each network is divided into multiple clusters and the nodes in the cluster is managed by the autonomous clustering. In the proposed scheme, each cluster in the networks autonomously

 ISBN: 978-1-61208-318-6

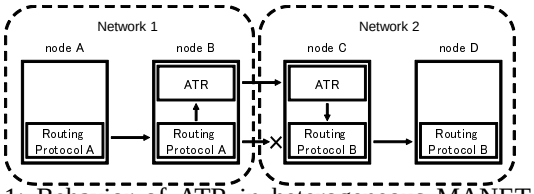


Figure 1: Behavior of ATR in heterogeneous MANET environment.

and dynamically selects one or more network gateway nodes from the nodes in the cluster. The nodes which become network gateway nodes can forward any packets to nodes of the different network by using the mechanism of ATR so that the interoperability between different networks can be provided.

A. ATR(Ad hoc Traversal Routing)

We have proposed ATR that provides the communication between different networks in heterogeneous MANETs. We define a node that both any routing protocol and ATR work as a NwGW (Network GateWay) node. A NwGW node converts from control packets of a routing protocol into control packets of ATR and vice versa. As a result, a node in a network can communicate with another node in a different network through NwGW nodes.

We explain ATR using an example as shown in Figure 1. Given that there are two networks, which are Network 1 and Network 2, and nodes A and B belong to Network 1 and nodes C and D belong to Network 2. When node A wants to communicate with node D, the route between nodes B and C cannot be created because the routing protocols are different. However, in this example, ATR works on nodes B and C so that nodes between Network 1 and Network 2 can communicate with each other through nodes B and C. Node B that receives a control packet of routing protocol A from node A converts from the control packet to a control packet of ATR, and then forwards it to node C. Node C that receives the control packet of ATR converts from the control packet to the corresponding control packet of routing protocol B, and then forwards it to node D. As a result, the route between nodes A and D can be created through nodes B and C.

B. Network Gateway Selection Scheme Based on Autonomous Clustering

In each network, the NwGW nodes must be selected to forward any packets from a network to another network in heterogeneous MANET environment. In MANETs, since all nodes are always moving, NwGW nodes must be dynamically selected from nodes in each network according to the topology change. Therefore, in order to select NwGW nodes dynamically in each MANET, we proposed an autonomous clustering-based dynamic network gateway selection for heterogeneous MANETs [7]. Autonomous clustering is the scheme to divide the network into multiple clusters. Each cluster consists of one cluster head, some gateways, and cluster members.

Figure 2 shows the outline of the heterogeneous MANET environment based on the network gateway selection scheme. In Figure 2, there are two networks, which are Networks 1 and 2. In each network, NwGW nodes are selected based

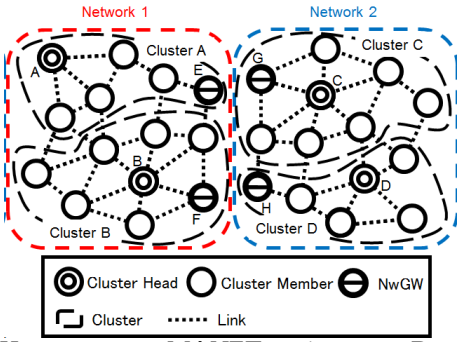


Figure 2: Heterogeneous MANET environment Based on Network Gateway Selection Scheme.

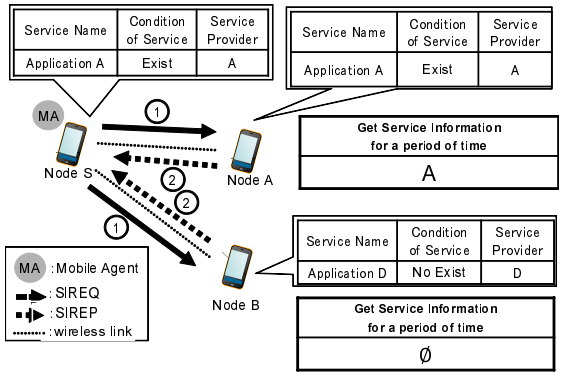


Figure 3: Mobile agent migration mechanism.

on the dynamic network gateway selection scheme. In this environment, any packets are forwarded from Network 1 to Network 2 through the link between two NwGW nodes in different networks, which are nodes F and H.

III. MOBILE AGENT MANAGEMENT MECHANISM

A. Outline

The purpose of the mobile agent-based service collection and dissemination scheme is to collect and disseminate service information for a shorter time in heterogeneous MANETs. In heterogeneous MANETs, the proposed scheme creates a mobile agent when a NwGW node holds service information and the other mobile agents do not come to the NwGW node for a certain period.

B. Mobile Agent Migration Mechanism

In Moving Average-based (MA-based) migration mechanism, mobile agents recognize the service information that each of neighboring nodes newly obtained for a specified period and migrate to the neighboring node that the number of the services is the lowest. Here, we define the specified period as the service collection time.

We explain the mobile agent migration mechanism using Figure 3. Given that there are three nodes: nodes S, A, and B, and a mobile agents is staying at node S in Figure 3. Each table denotes the list of service information that each node manages. First, as shown in Figure 3①, node S that a mobile agent is staying broadcasts a Service Information REQuest (SIREQ) including Service ID of node S to obtain the service

 ISBN: 978-1-61208-318-6

information from neighboring nodes like SN-based migration mechanism. Then, as shown in Figure 3②, nodes A and B that received the SIREQ from node S sends a Service Information REPly (SIREP) back to node S. Here, the SIREP includes the service information that the mobile agent does not have and each node (that is, nodes A and B) has, and a set of services (SPT_i) that each node i newly obtained for a specified period. In this time, if the neighboring nodes do not have the service information contained in the SIREQ, they can obtain a new service information from the SIREQ. Finally, the mobile agent that received the SIREPs from the neighboring nodes migrates to the node that the number of new obtained services for the specified period is the lowest.

For example, given that node S receives SIREPs from nodes A and B as shown in Figure 3. In this case, $SPT_A = \{A\}$ and $SPT_B = \emptyset$. Therefore, the mobile agent migrates to node B such that $|SPT_B| < |SPT_A|$.

C. Mobile Agent Creation-Termination Mechanism

The mobile agent creation-termination mechanism creates or terminates mobile agents to control the number of mobile agents in the network.

1) Mobile Agent Creation Conditions: A mobile agent is created on a node when one of three conditions is satisfied.

Condition 1: A node with a service generates or updates a service.

Condition 2: A node does not receive a SIREQ for a specified period.

Condition 3: A node is selected as a NwGW node and it has collected service information for a specified period, which is defined as a service collection time, in the past.

By Condition 1, the generated service or updated service is disseminated by mobile agents. By Condition 2, the node creates a mobile agent when each node judges that the number of mobile agents in the network is low. By Condition 3, a NwGW node creates the mobile agent in order to easily migrate the mobile agent between different networks.

2) Mobile Agent Termination Conditions: A mobile agent is terminated when one of following two conditions is satisfied.

Condition 1: A mobile agent receives a SIREQ from the other mobile agents.

Condition 2: A node with a mobile agent does not have neighboring nodes.

By Condition 1, the node terminates a mobile agent when the number of mobile agents in the network is high. By Condition 2, the node terminates a mobile agent when a mobile agent cannot disseminate service information.

D. Types of Mobile Agents

The proposed scheme uses two types of mobile agents in heterogeneous MANETs.

IMA (Internal MA): A mobile agent disseminates service information in a network.

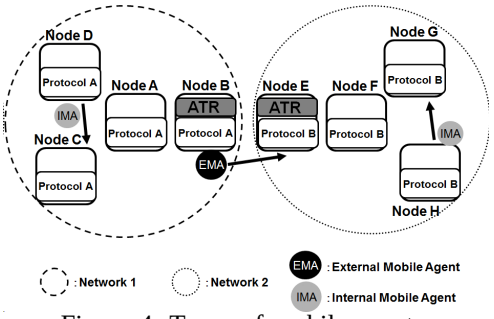


Figure 4: Types of mobile agents.

TABLE I: Simulation environment in Simulation I.

Network Simulator	QualNet ver. 5.0
Field size [m^2]	1570×1570
Number of nodes	200
Number of networks	2
Number of nodes in each network	100
Transmission range [m]	250
MAC protocol	IEEE802.11b
Node moving speed [m/s]	1∼4
Node mobility model	Random Way Point
Size of MAs [byte]	4096

EMA (External MA): A mobile agent disseminates service information between different networks.

IMAs are created by Conditions 1 and 2 of the mobile agent creation mechanism in order to disseminate service information in a network. EMAs are created by Condition 3 of the mobile agent creation mechanism in order to disseminate service information between NwGW nodes in each network. A mobile agent transits from the EMA to the IMA in order to disseminate service information in the network after the EMA has migrated between different networks.

In Figure 4, there are two networks, which are Networks 1 and 2 in heterogeneous MANET environment, and nodes B and E are NwGW nodes in Networks 1 and 2, respectively. In the proposed scheme, the IMA migrates from a node to another node while collecting and disseminating service information only in each network. On the contrary, the EMA is generated at node B in Network 1, and then it migrates from node B to node E to disseminate service information in Network 2. After that, the EMA becomes the IMA of Network 2 and disseminates service information only in Network 2.

IV. Simulation Evaluation

We conducted simulation experiments to evaluate the proposed scheme through network simulator QualNet [8]. We show the effectiveness of the proposed scheme in comparison with Random scheme.

Random scheme is a scheme to randomly migrate mobile agents from a node to another node without considering the efficient service collection and dissemination. Here, mobile agents in Random scheme are generated only by Conditions 1 and 2 of mobile agent creation conditions.

A. Simulation I

In Simulation I, we investigate the property of the service collection time of the proposed scheme.

 ISBN: 978-1-61208-318-6

TABLE II: Service dissemination time versus service collection time in Simulation I.

Dissemination Rate[%]	Service collection time [s]							
	Case 1				Case 2			
	10	50	100	200	10	50	100	200
60	115	96	86	87	102	89	84	84
70	138	113	102	102	121	105	99	99
80	165	133	121	121	147	125	118	118
90	207	163	148	148	187	154	146	146

TABLE III: Total number of EMAs in Simulation I.

Service collection time [sec]	10	50	100	200
Case 1	7	35	63	88
Case 2	14	48	73	91

1) Simulation Plan: We conducted simulation experiments in the environment where there are two types of MANETs in the field. Table I shows simulation environment. In the simulation experiment, the service collection time of the mobile agent creation condition is set at 10, 50, 100, and 200 seconds. Evaluation criteria are the service dissemination time versus the service dissemination ratio and the number of EMAs. Here, the service dissemination ratio is the ratio of the number of service disseminated nodes to the number of nodes. In addition, one node in Network 1 generates a new service at fixed intervals from the simulation start to the end. First, we investigate the influence on the service generation interval in two cases as follows. In Case 1, the number of services and the service generation interval are set at 10 and 50 seconds, while in Case 2, the number of services and the service generation interval are set at 100 and 5 seconds, respectively. Next, we show the effectiveness of the proposed scheme in comparison with Random scheme.

2) Results for the effect on the service collection time: Table II shows the service dissemination time versus the service collection time in Cases 1 and 2, and Table III shows the number of generated EMAs. The service dissemination times in both Cases 1 and 2 become shorter as the service collection time increases from 10 to 100, while there is no difference between 100 and 200 of service collection time. As shown in Table III, the number of EMAs increase as the service collection time becomes more. When the service collection time is 200, the number of EMAs becomes more than the service collection time 100. However, the dissemination time of the service collection time 200 does not become shorter than that of the service collection time 100. As a result, even if the service collection time becomes more and more EMAs are generated, the service dissemination time does not become shorter. Therefore, we can confirm that the service collection time 100 is appropriate for the proposed scheme in this experiments.

3) Results for the effectiveness of the proposed scheme: In the simulation experiments, the service collection time of the proposed scheme is set at 100 [sec]. Table IV shows the service dissemination time in Cases 1 and 2. The proposed scheme becomes shorter than Random scheme in both cases because of the mobile agent migration mechanism as well as the mobile agent creation condition 3.

Next, we focus on the service dissemination time in Network 1 where there is the service generation node. Table V shows the service dissemination time in Network 1. As shown in Table V, the dissemination time of the proposed scheme becomes shorter than that of Random scheme. In the proposed scheme, mobile agents migrate to a node in which the number of service information obtained for the service collection time is the lowest, while in Random scheme, mobile agents randomly select the node to which they migrate.

TABLE IV: Service dissemination time [sec] in Simulation I.

Dissemination Rate [%]	Case 1		Case 2	
	Proposed	Random	Proposed	Random
60	86	137	84	127
70	102	164	99	154
80	121	199	118	189
90	148	252	146	242

TABLE V: Service dissemination time [sec] in Network 1 in Simulation I.

Dissemination Rate [%]	Case 1		Case 2	
	Proposed	Random	Proposed	Random
60	62	73	59	68
70	76	90	74	86
80	94	109	92	107
90	121	139	118	137

Next, we focus on the time when the service information is forward from Network 1 to Network 2 in order to show the effectiveness of the mobile agent creation condition 3. Table VI shows the service dissemination start time in Network 2. As shown in Table VI, the proposed scheme can disseminate the service information to the neighboring network (Network 2) in a shorter time. In the proposed scheme, in case that NwGW nodes have obtained a new service information for the service collection time in the past, they create EMAs to disseminate the service information to the neighboring network. On the contrary, in Random scheme, only when mobile agents arrive at NwGW nodes, the service information is disseminated to the neighboring network. Therefore, the proposed scheme can provide the efficient service information dissemination among networks.

Finally, we focus on the overhead. Figure 5 and Table VII show the number of MAs versus simulation time and the number of MAs per second. The proposed scheme creates EMAs by the mobile agent creation condition 3 to disseminate the service information to the neighboring network. However, as shown in Table VII, there is no difference between the proposed scheme and Random scheme because the number of MAs is adjusted by the mobile agent creation-termination mechanism.

Table VIII shows the total volume of control packets from the simulation start to 1000 seconds. As shown in Table VIII, the proposed scheme becomes 10 percents more than Random scheme because SIREQ packets are generated by the mobile agent termination mechanism due to EMAs.

B. Simulation II

In Simulation II, we confirm that the proposed scheme can be applicable for service collection and dissemination in heterogeneous MANET environment where the number of networks is three. From the results of Simulation I, the service collection time is set at 100 seconds in Simulation II.

 ISBN: 978-1-61208-318-6

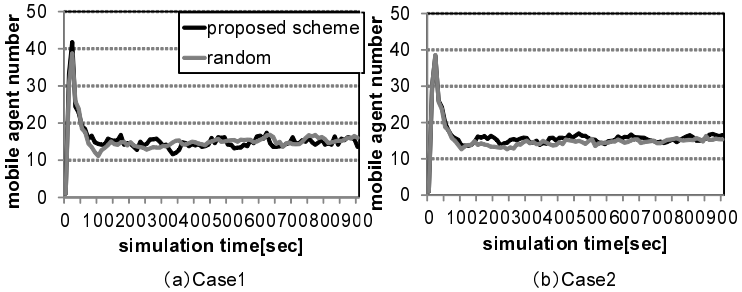


Figure 5: Number of MAs versus simulation time in Simulation I.

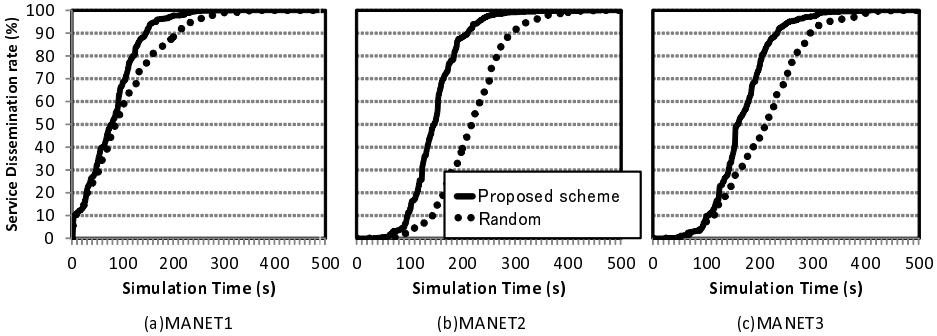


Figure 7: Service dissemination rate versus simulation time in each MANET in Simulation II.

TABLE VI: Service dissemination start time [sec] in different network in Simulation I.

Case 1		Case 2	
Proposed	Random	Proposed	Random
32	102	32	84

TABLE VII: Average number of MAs per second in Simulation I.

Case 1		Case 2	
Proposed	Random	Proposed	Random
15.3	15.4	15.9	15.1

1) Simulation plan: Table IX shows the simulation environment in Simulation II. Three types of networks, which are MANETs 1, 2, and 3, coexist in the field and we conduct simulation experiments in the heterogeneous MANET environment. The total number of nodes in the network is 300 and the number of nodes in each MANET is 100. The node moving speed is between 1 and 4 m/s. In this environment, the number of network gateway nodes (NwGW nodes) became 5.5 on average. In addition, a node generates a mobile agent at 100

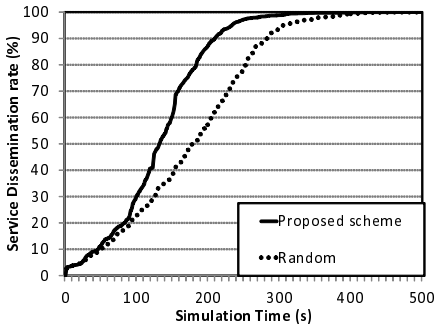


Figure 6: Service dissemination rate versus simulation time.

TABLE VIII: Total control overhead [Mbyte] in Simulation I.

Case 1		Case 2	
Proposed	Random	Proposed	Random
5.5	5.1	22.6	19.7

TABLE IX: Simulation environment in Simulation II.

Simulator	Qualnet ver. 5.0
Field size [m^2]	1570 $\times$ 1570
Number of nodes	300
Number of networks	3
Number of nodes in each network	100
Node moving speed [m/s]	1$\sim$4
Node mobility model	Random Way Point
Number of generated services	10
Transmission range [m]	250
MAC protocol	IEEE802.11b
Size of MAs [byte]	4096

second interval by Mobile agent creation condition 2, and when NwGW node that holds any service information does not hold a mobile agent for 30 seconds, it generates a mobile agent by Mobile agent creation condition 3. In the simulation, nodes in MANET 1 generates ten services at 10 second interval from the simulation start.

2) Simulation results: Figure 6 shows the service dissemination rates of the proposed scheme and Random.

From Figure 6 , we can confirm that the proposed scheme can disseminate service information in a shorter time than Random. This is because mobile agents can appropriately decide the next node that they should move according to the mobile agent migration mechanism. On the other hand, in Random scheme mobile agents move to the node that does not require the service dissemination and collection because mobile agents randomly move around the network. In addition, the proposed scheme generates EMAs (External Mobile Agents) to forward service information between different MANETs. Therefore, in

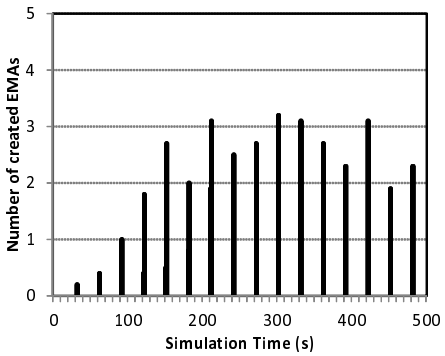


Figure 8: Number of generated EMAs versus simulation time in Simulation II.

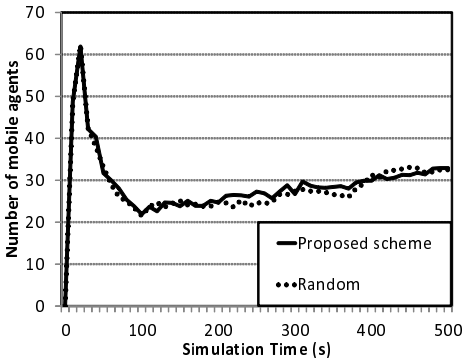


Figure 9: Number of mobile agents versus simulation time in Simulation II.

the heterogeneous MANET environment, the proposed scheme could disseminate service information from MANET 1 to MANETs 2 and 3 in a shorter time.

Figures 7 and 8 show the service dissemination rate in each MANET and the number of generated EMAs, respectively. From Figure 7, in each MANET, the proposed scheme can disseminate service information in a shorter time than Random. Figure 7(a) shows the service dissemination rate in MANET 1. In MANET 1, the services are generated and disseminated by mobile agents, and then the proposed scheme can disseminate service information in a shorter time than Random. In the proposed scheme, mobile agents migrate from a node to another node by the mobile agent migration mechanism, while in Random scheme, they migrate randomly. On the contrary, because MANETs 2 and 3 does not generate service information, MANETs 2 and 3 can disseminate service information after nodes in MANETs 2 and 3 receive service information from mobile agents that are migrated from MANET 1. Figures 7(b) and 7(c) show the service dissemination rate in MANETs 2 and 3. As shown in Figure 8, the proposed scheme can disseminate service information between different MANETs by EMAs. Therefore, especially in MANETs 2 and 3, the proposed scheme can disseminate service information in a shorter time than Random in comparison with MANET 1. Consequently, we can say that the proposed scheme can disseminate service information in a shorter time in the heterogeneous MANET environment.

Next, we focus on the overhead. It is expected that the proposed scheme becomes higher overhead than Random because the proposed scheme generates EMA for service dissemination and collection. Figure 9 shows the number of mobile agents in the proposed scheme and Random. We can confirm that there are no big difference between the proposed scheme and Random from Figure 9. In addition, Table X shows the average number of mobile agents per 1 second in the proposed scheme and Random. Here, in the proposed scheme, the ratio of EMAs to MAs is only 0.003 %. As shown in Table X, the difference between the proposed scheme and Random is very small. This is because the total number of mobile agents is controlled by the mobile agent creation-termination mechanism. As a result, we can say that that the proposed scheme is appropriate for the service dissemination and collection in heterogeneous MANET environment.

TABLE X: Average number of mobile agents in Simulation II.

	Proposed scheme	Random
Average number of MAs [/1 sec]	28.8	28.3

V. Conclusion

We have proposed a mobile agent-based service information collection and dissemination in heterogeneous MANETs, and its node architecture, and shown the effectiveness of the scheme based on the new node architecture in terms of the service dissemination time through simulation experiments. In the future work, we are planning to implement the proposed node architecture on the mobile terminals like android smart phones and verify the behavior in heterogeneous MANET environment through field experiments.

Acknowledgments

This work was supported in part by JSPS KAKENHI (Grant Number 24700073 and 24300028), the Hiroshima City University under Grant for Special Academic Research, and MIC SCOPE (No.131408006).

References

[1] C. K. Toh, “Ad Hoc Mobile Wireless Networks Protocols and Systems,” Prentice Hall Inc., 2002.

[2] C.N. Ververidis and G. C. Polyzos, “Service discovery for mobile ad hoc networks: A survey of issues and techniques,” Communications Surveys and Tutorials, IEEE Computer Society, pp.30–45, 2008.

[3] C. Cho and Duckki Lee, “Survey of Service Discovery Architectures for Mobile Ad hoc Networks,” Department of Computer and information sciences and Engineering, pp.30-45, 2005.

[4] T. Ohta, S. Inoue, and Y. Kakuda, ”An adaptive multihop clustering scheme for highly mobile ad hoc networks,” Proc. 6th IEEE Int’l Symp. on Autonomous Decentralized Systems (ISADS2003), pp.293-300, April 2003.

[5] T. Hashimoto, T. Ohta, and Y. Kakuda, “Evaluation of mobile agent-based service dissemination schemes in MANETs,” Proc. 2nd Int’l Conf. on Networking and Computing (ICNC’11), pp.257–260, Nov. 2011.

[6] S. Fujiwara, T. Ohta, and Y. Kakuda, “An inter-domain routing for heterogeneous mobile ad hoc networks using packet conversion and address sharing,” Proc. 32th IEEE Int’l Conf. on Distributed Computing Systems Workshops (ADSN 2012), pp.349–355. June 2012.

[7] K.Okano, T.Ohta, and Y.Kakuda, ”A dynamic network gateway selection scheme based on autonomous clustering for heterogeneous mobile ad hoc network environment”, Proc. 7th IEEE International Workshop on Heterogeneous, Multi-Hop, Wireless and Mobile Networks (GC’12 Workshop - HeterWMN 2012), pp.513-517, Dec. 2012.

[8] Scalable Network Technologies, Inc., ”QualNet network simulator”. http://www.scalable-networks.com/, (accessed 2013-12-29).

 ISBN: 978-1-61208-318-6

A Bio-Inspired Transmit Power Control Algorithm for Linear Multi-Hop Wireless Networks

Hyun-Ho Choi
Dept. of Electrical, Electronic and Control Engineering
Hankyong National University
Republic of Korea
Email: hhchoi@hknu.ac.kr

Jung-Ryun Lee
School of Electrical Engineering
Chung-Ang University
Republic of Korea
Email: jrlee@cau.ac.kr

***Abstract*—Inspired by the flocking behavior, we propose a distributed transmit power control (TPC) algorithm for maximizing the end-to-end rate in a linear multi-hop wireless network. As each bird in flock goes to match its velocity with the average velocity of its neighbors in a distributed manner, each node on the multi-hop path matches its transmission rate with the average transmission rate of its neighbor nodes by controlling its transmit power. We verify that this TPC algorithm performing a local rate-average strategy maximizes the end-to-end rate of the wireless multi-hop link. Simulation results show that as with the flocking algorithm, the proposed TPC algorithm enables all link rates to converge to the same value, and also significantly decreases the power consumption of multi-hop nodes, while maximizing the multi-hop end-to-end rate.**

Keywords-transmit power control; bio-inspired algorithm; flocking algorithm; end-to-end rate maximization; multi-hop network

I. INTRODUCTION

In multi-hop networks, an increase in the number of hops improves each link budget, but generates more traffic in the network. This eventually increases the access collision and interference levels and so degrades the multi-hop end-to-end performance [1]. The most effective method for breaking through this trade-off is a transmit power control (TPC) that controls the transmit power of multi-hop nodes in order to mitigate the strong interference while ensuring a reasonable link budget [2].

The typical TPC algorithms in wireless multi-hop networks are mainly based on the condition of individual links [3]-[10]. In [3]-[5], the minimum transmit power level is used to guarantee the signal-to-interference plus noise ratio (SINR) required at the receiving node depending on the quality of service (QoS) of the transmitted packets, aiming at achieving not only interference mitigation but also power saving. In [6] and [7], the transmit power is controlled based on the packet size. The greater is the packet size, the higher is the packet error rate (PER); the transmit power increases with an increase in the packet size. In [8], the transmit power is determined based on the channel state information (CSI) to maintain a constant bit error rate (BER) at the receiver. In [9], the transmit power for control packets increases to prevent interference from hidden nodes in the IEEE 802.11 system. In [10], a common power level is determined from the perspective of the overall network capacity in order to guarantee the rates of bi-directional links.

The objective of typical TPC algorithms in wireless multi-hop networks is mostly to minimize the transmit power consumption while ensuring the given QoS (i.e., SINR or BER) of each individual link on the multi-hop path [2]. Such a power minimization problem subject to the constant SINR requirement can guarantee the required end-to-end rate of a multi-hop link, but cannot maximize it, because the achievable maximum SINR value that maximizes the end-to-end rate of a given multi-hop link is unknown. This achievable maximum SINR value varies depending on the transmit power of each multi-hop node due to the mutual interference among wireless links. Therefore, the transmit powers of all nodes should be considered jointly to maximize the end-to-end rate in a given inter-link interference situation. However, this joint TPC operation requires a complex calculation and causes a significant overhead for sharing information among all nodes to determine the optimal transmission power in each node [1].

To solve such a complex optimization problem, in this paper, we pay attention to a biological system known as the *flocking behavior*, which is exhibited when a group of birds, called a flock, are foraging or in flight. Flocks behave with a very simple rule in complex, unstructured, and dynamically changing environments, but they show a emergent behavior achieving their common goal robustly and efficiently. By understanding the similarities between the flocking behavior and the wireless multi-hop transmission, we adapt the underlying principles of the flocking behavior to the TPC algorithm in wireless multi-hop networks. As the flocking algorithm operates in simple and distributed manners and shows converged phenomena, the proposed bio-inspired TPC (BiTPC) algorithm follows a low-complex operation without a centralized controller and converges to maximize the end-to-end rate in the considered linear multi-hop network.

The rest of this paper is organized as follows. In Section II, we describe the optimization problem for maximizing the end-to-end rate in the linear multi-hop wireless network. In Section III, we introduce the flocking behavior and mention its properties. In Section IV, we explain the proposed BiTPC algorithm in detail. In Section V, we investigate the optimality of the proposed BiTPC algorithm. In Section VI, we discuss

 ISBN: 978-1-61208-318-6

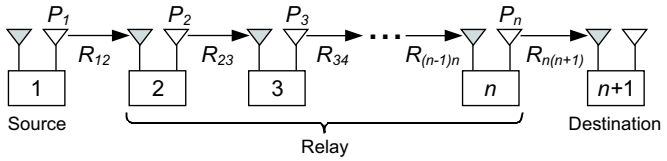


Figure 1. Considered linear multi-hop wireless network.

the simulation results. Finally, we conclude the paper in Section VII.

II. Problem Description

Fig. 1 shows a considered linear multi-hop wireless network consisting of n hops from a source to a destination. Here, we define some notation, as follows:

- P_i: transmit power of node i
- N_i: noise power at node i
- I_i: interference power received at node i
- g_{ij}: channel gain from node i to node j
- SINR$_{ij}$: SINR from node i to node j
- R_{ij}: link rate from node i to node j

By max-flow min-cut theorem, the end-to-end rate from the source node to the destination node is determined by the smallest value of the link rates in the multi-hop [11]. Therefore, it is defined as

$$R_{e2e} := \min\{R_{12}, R_{23}, \cdots, R_{n(n+1)}\}. \tag{1}$$

Our objective is to obtain the transmit powers of all the transmitting nodes that maximize the end-to-end rate R_{e2e}. This is described as the following optimization problem:

$$\max_{\mathbf{P}} R_{e2e} = \max_{\mathbf{P}} \min\{R_{12}, R_{23}, \cdots, R_{n(n+1)}\} \tag{2}$$

$$\text{s.t.} \quad \mathbf{P} = [P_1 P_2 \cdots P_n] \tag{3}$$

$$R_{ij} \le \log_2(1+\text{SINR}_{ij}) \text{ for } i=1,2,\cdots,n \text{ and } j=i+1$$
$$= \log_2\left(1+\frac{P_i g_{ij}}{I_j+N_j}\right)$$
$$= \log_2\left(1+\frac{P_i g_{ij}}{\sum_{\forall k\neq i,j} P_k g_{kj}+N_j}\right) \text{[b/s/Hz]} \tag{4}$$

$$P_i \le P_{max} \quad \text{for } i=1,2,\cdots,n \tag{5}$$

where $\mathbf{P}$ is the vector of the transmit power of each node, P_{max} is the maximum transmit power constraint, and I_j is given by $\sum_{\forall k\neq i,j} P_k g_{kj}$ as the sum of interferences from other transmitting nodes that use the same radio resource. Here, we assume that all nodes use the same resource without considering a particular scheduling because our main purpose is to show the applicability of the proposed BiTPC algorithm under the inter-link interference condition. In this context, we also assume that the relay node is a full-duplex relay; therefore, it is possible to receive and transmit packets simultaneously without self-interference and processing delay [12]. It should be noted that the considered optimization problem is complicated to solve directly (i.e., it is not convex) because the transmit power of each node affects all link rates.

III. Flocking Behavior

Flocking represents the phenomenon in which self-propelled individuals organize into an ordered motion by using only limited environmental information and simple rules. For example, a flock of birds whose members are moving in $\mathbb{R}^3$ shows that the state of the flock converges to one in which all birds fly with the same velocity. The simple rule of this flocking behavior is known that each bird autonomously adjusts its velocity according to the velocities of its neighbors. The recent representative Cucker-Smale flocking model [13] explains the flocking behavior, that is, at time t and for bird i, every bird adjusts its velocity v_i as follows:

$$v_i(t+1) - v_i(t) = \frac{\lambda}{N}\sum_{j=1}^{N}\psi(|x_j - x_i|)(v_j - v_i) \tag{6}$$

where N is the number of birds, λ is a coupling strength as a learning parameter, and x_i is the position of bird i. ψ means a communication range function which can be set to $\psi(|x_j - x_i|) = 1$ only in case of $|x_j - x_i| \le r$. So, this flocking rule can be interpreted as a local averaging algorithm for bird velocity. In addition, Cucker-Smale flocking model ensures that an interacting N-particle system $\{(x_i(t), v_i(t))\}_{i=1}^{N}$ has time-asymptotic convergence properties, as follows: [13], [14]

1) Velocity alignment:

$$\lim_{t\to\infty} |v_i(t) - v_j(t)| = 0, \quad \text{for } \forall i \neq j. \tag{7}$$

2) Formation of a group:

$$\sup_{0\le t<\infty} |x_i(t) - x_j(t)| < \infty, \quad \text{for } \forall i \neq j. \tag{8}$$

Then, why do birds flock together? There are a variety of reasons, such as foraging, mating, navigation, protection from predators, etc. Among them, one theory is that in coordinated flight of birds, there is an aerodynamic advantage to flying behind [15]. As a front bird moves its wings up and down, a strong current of air is created and flows backward. This moving wave of air uplifts the bird behind it. That is, each bird flying ahead creates an air wave that helps the bird flying behind it. This cooperation reduces the energy consumption of birds and thus allows them to arrive faster at their destination [16]. Because of these aerodynamic interactions, the best way for a bird group to arrive at the destination as soon as possible without straggling (i.e., to maximize the minimum bird speed) is to cooperate with each other in a way that the speedy birds fly in front of the tardy birds. As a result, this cooperation makes all birds fly at the same velocity. Therefore, it is noticed that the flocking algorithm equalizing all the birds' velocity can be an appropriate solution to achieve the goal of bird flock.

IV. Proposed Bio-Inspired Transmit Power Control Algorithm

Both the multi-hop transmission and the flocking behavior have the objective function of the max-min type. Moreover, the flocking algorithm is a basically simple and distributed approach and has the convergence properties. Considering these similarity and advantages, we adapt the flocking algorithm to the TPC algorithm for the wireless multi-hop transmission.

 ISBN: 978-1-61208-318-6

Similar to the flocking algorithm, we equalize the link rates by adjusting the transmit power of each node. That is to say, the proposed BiTPC algorithm controls the transmit power of each node in such a way to equalize all the link rates (i.e., $R_{12} = R_{23} = \cdots = R_{n(n+1)}$). Because the rate of link ij, R_{ij}, is related to the transmit powers of all the other nodes ($P_k, \forall k \neq i, j$) as well as to the transmit power of node i (P_i), the control of one node's transmit power influences all the other nodes' rates, and this requires an iteration to obtain the final equal rate value. At each step, each node recognizes the rate value of its neighboring nodes and uses the average of the recognized rate values as its next target rate, as each bird in the flock matches its velocity with its neighboring birds repeatedly in a distributed manner. Thereafter, each transmitting node decides the transmit power to achieve the target rate individually. This distributed local rate-average operation is repeated until all the link rates converge to the same value.

The flow chart of the proposed BiTPC algorithm is shown in Fig. 2, and its detailed operation follows these steps:

1) All the transmitting nodes set the initial transmit power to the maximum transmit power P_{max}.
2) The transmitting node i sends the packet to its receiving node j by using the transmit power $P_i(t)$ decided for time t.
3) Upon receiving the packet, the receiving node j measures its SINR_{ij} and feeds it back to its transmitting node i.
4) On the basis of the SINR feedback, the transmitting node i calculates its current link rate as $R_{ij}(t) = \log_2(1 + \text{SINR}_{ij}(t))$.
5) Each transmitting node shares the information of $R_{ij}(t)$ or $\text{SINR}_{ij}(t)$ with its neighboring nodes. Note that the rate and SINR can be converted to each other. As a candidate sharing method, the overhearing technique is possible [17]. With this technique, the node overhears the SINR feedback or the transmitted modulation and coding set (MCS) information of the adjacent nodes; therefore, these information of adjacent links can be shared among nodes without additional signalling for sharing.
6) The next target rate $R_{ij}(t+1)$ is determined as the average value of the recognized adjacent link rates, as follows:

$$R_{ij}(t+1) - R_{ij}(t) = \frac{1}{\eta} \sum_{\forall k, l=k+1} \psi(|x_k - x_i|)\,(R_{kl}(t) - R_{ij}(t)) \quad (9)$$

$$\Rightarrow R_{ij}(t+1) = \frac{1}{\eta} \sum_{kl \in \{\text{neighbor links}\}} R_{kl}(t) \quad (10)$$

where η is the total number of neighbor links whose rate information is shared. The communication range function $\psi(|x_k - x_i|) = 1$ if the node k is the neighbor of the node i and the node k's rate information is shared. Otherwise, $\psi(|x_k - x_i|) = 0$.
7) If the next target rate $R_{ij}(t+1)$ has little difference with the current target rate $R_{ij}(t)$ (i.e., $R_{ij}(t+1) - R_{ij}(t) \leq \epsilon$ where $\epsilon > 0$ is small enough), $P_i(t)$ is determined to be the final transmit power and the iteration ends. Otherwise, from (4) and (5), the next transmit power $P_i(t+1)$ is calculated to obtain the next target rate $R_{ij}(t+1)$, as follows:

$$P_i(t+1) = \min\left[\frac{\{2^{R_{ij}(t+1)} - 1\}\{I_j(t) + N_j(t)\}}{g_{ij}}, P_{max}\right] \quad (11)$$

where $\frac{I_j(t)+N_j(t)}{g_{ij}}$ can be derived from the $\text{SINR}_{ij}(t)$ feedback. Thereafter, the operation continues at Step 2.

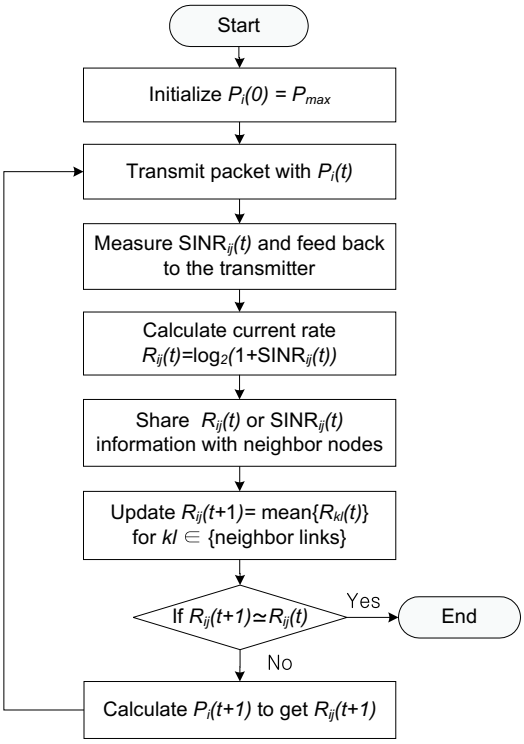


Figure 2. Flow chart of proposed BiTPC algorithm.

V. Proof of Optimality

The rate set of wireless links that use the same radio resource at the same time affect each other owing to their mutual interference [18]. Therefore, as expressed in (4), it is always possible to increase the rate of one link at the expense of another. We call this property *solidarity* and define it as follow.

Definition 1 (Solidarity Property): A subset $\mathcal{X}$ of $\mathbb{R}^n$ has a solidarity property if and only if for all $i \in \{1, 2, \cdots, n\}$, for all $\mathbf{x} \in \mathcal{X}$ where the i-th element $\mathrm{x}_i > 0$, and for all $0 < \alpha_i < \epsilon$ where $\epsilon > 0$ is small enough, the variation of x_i, $\mathrm{x}_i \pm \alpha_i$, induces variations of the other elements, $\mathrm{x}_j \mp \alpha_j$ for $\forall j \neq i$ and $0 < \alpha_j < \epsilon$, and the changed vector $\mathbf{y} = \mathbf{x} \pm \alpha_i \mathbf{e}_i \mp \sum_{\forall j \neq i} \alpha_j \mathbf{e}_j$ where $\mathbf{e}_i$ is a unit vector still belongs to $\mathcal{X}$.

According to the definition of the solidarity property, we state the following proposition and prove it in order to verify that the proposed BiTPC algorithm is an optimal solution to maximize the end-to-end rate of the wireless multi-hop link.

Proposition 1: If a set $\mathcal{X}$ has the solidarity property, then the *max-min fair* vector $\mathbf{x} \in \mathcal{X}$ has all components equal,

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Distance btw. source and destination	1000 m
Minimum distance between nodes	10 m
Number of transmission hops	1∼15
Information-sharing range	2 hops (default)
Maximum transmit power	23 dBm
Distance-dependent path loss	$128.1+37.6\log_{10}R$ [dB], R in km
Noise figure	9 dB
Threshold for convergence check	10^{-2}

that is, $\mathrm{x}_i = \mathrm{x}_j$ for $\forall i, j$ when the minimum value of $\mathbf{x}$ is maximized.

Proof: Suppose the contrary that there exists a max-min fair vector $\mathbf{x}$, such that $\mathrm{x}_i \neq \mathrm{x}_j$ for some $i \neq j$ on $\mathcal{X}$ with the solidarity property. Let x_i be the largest component of $\mathbf{x}$. Then, for sufficiently small ϵ such that $0 < \epsilon < \min_j\{\mathrm{x}_i - \mathrm{x}_j\}$, we have

$$\mathrm{x}_i > \mathrm{x}_j + \epsilon \quad \text{for } \forall j \neq i. \tag{12}$$

According to the definition of the solidarity property, for $0 < \alpha_i, \alpha_j < \epsilon$, we can find another vector $\mathbf{y} \in \mathcal{X}$ such that

$$\mathbf{y} = \mathbf{x} - \alpha_i \mathbf{e}_i + \sum_{\forall j \neq i} \alpha_j \mathbf{e}_j. \tag{13}$$

That is, $\mathrm{y}_i = \mathrm{x}_i - \alpha_i$ and $\mathrm{y}_j = \mathrm{x}_j + \alpha_j$ for $\forall j \neq i$. This satisfies $\mathrm{y}_i > \mathrm{x}_i - \epsilon > \mathrm{x}_j$ and $\mathrm{y}_j > \mathrm{x}_j$ for $\forall j \neq i$. Therefore, all elements of $\mathbf{y}$ are greater than x_j, i.e.,

$$\max\{\min(\mathbf{y})\} > \max\{\min(\mathbf{x})\} = \mathrm{x}_j \tag{14}$$

which contradicts the supposition that $\mathbf{x}$ is the max-min fair vector. ■

The rate set of links consisting of the wireless multi-hop link has the solidarity property. Moreover, the multi-hop end-to-end rate is determined by the minimum link rate. Consequently, from Proposition 1, the multi-hop end-to-end rate is maximized when all the link rates are equal. Therefore, the proposed BiTPC algorithm, which controls the transmit power of each node in such a way as to equalize all the link rates, maximizes the end-to-end rate of the multi-hop link.

VI. RESULTS AND DISCUSSION

We consider a one-way linear multi-hop wireless network, as shown in Fig. 1. Table I shows the simulation parameters. The distance between the source node and the destination node is fixed as 1000 m and the number of transmission hops is varied by controlling the number of relay nodes between them. The relay node is deployed randomly on the line connecting the source node with the destination node, and the requirement of the minimum distance between the nodes is 10 m. The default information-sharing range is 2 hops, within which the nodes can share their rate or the SINR value. The maximum power is set to 23 dBm and the path loss follows the 3GPP evaluation parameter [19]. For comparison, we consider a scheme using the maximum equal power without TPC and the SINR-based TPC algorithm with several target SINR values [3]-[5].

Fig. 3 shows the rate of each link and the transmit power of each node according to the iteration of the proposed algorithm at one topology. Here, we assume that the iteration process is stopped if the Euclidean length of a transmission power

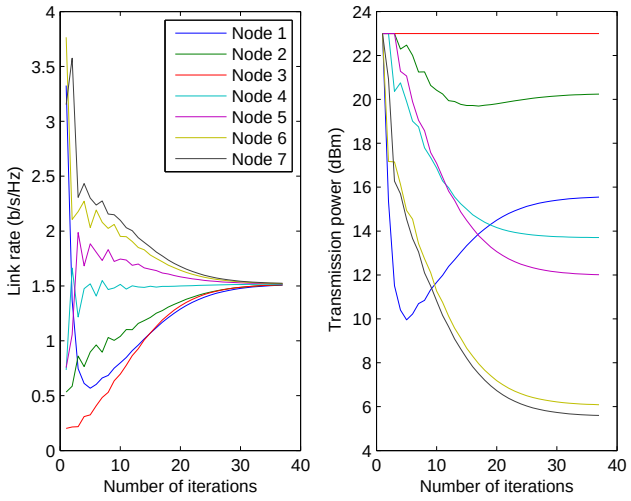


Figure 3. Link rate and transmission power vs. number of iterations.

vector $\mathbf{P}$ (i.e., norm of $\mathbf{P}$) is less than 10^{-2}. Note that this convergence check corresponds to a tight condition because in practical systems, set of possible bit rate is determined by several MCS levels. Therefore, the convergence speed can be improved further by relaxing the convergence condition especially in case that the channel variation and node mobility become more dynamic. As the iteration proceeds, the link rates converge, but the transmit powers become different and bounded. From the perspective of convergence, the simulation results have showed that $\lim_{t\to\infty}|R_i(t) - R_j(t)| = 0$ for $\forall i \neq j$ and $\sup_{0<t<\infty}|P_i(t) - P_j(t)| < P_{max}$ for $\forall i \neq j$. This is caused by the fact that the proposed BiTPC algorithm limits the maximum transmission power. Accordingly, the BiTPC algorithm shows the similar convergence properties to the flocking algorithm, as shown in (7) and (8).

Upon convergence, the node that initially had the best link rate (i.e., node 7) shows the lowest transmit power, and the node that initially had the worst link rate (i.e., node 3) maintains the maximum transmit power. That is, the nodes with good link quality reduce their transmit power, but the nodes with bad link quality maintain or slightly reduce their transmit power, in order to equalize all link rates. It should be noted that the final transmit power is inversely proportional to the initial link rate.

Fig. 4 shows the number of iterations needed for convergence, according to the number of transmission hops and the number of sharing hops. As the number of hops increases, the number of iterations increases exponentially because an increase in the number of nodes means that more time is required to equalize all the link rates. Moreover, as the number of sharing hops increases, the convergence becomes faster. This is because the increase in the number of sharing hops offers more adjacent link rates for averaging.

Fig. 5 shows the performance of the multi-hop end-to-end rate versus the number of transmission hops. The proposed BiTPC algorithm outperforms the scheme using the maximum equal power without TPC and the SINR-based TPC with a target SINR (γ) fixed at 0, 3, or 10 dBm. This is because

 ISBN: 978-1-61208-318-6

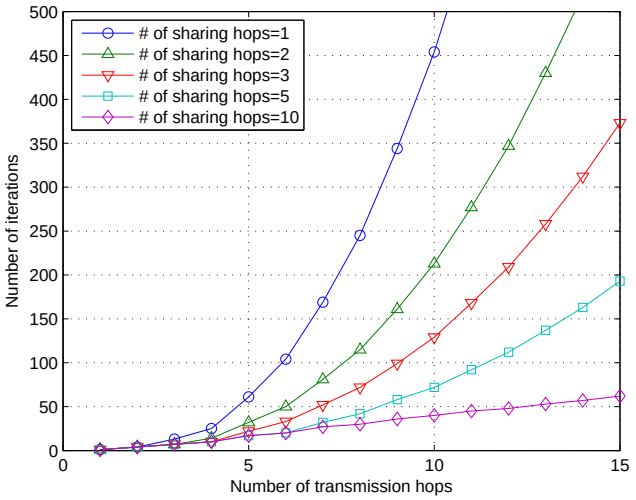


Figure 4. Number of iterations vs. number of hops and number of sharing hops.

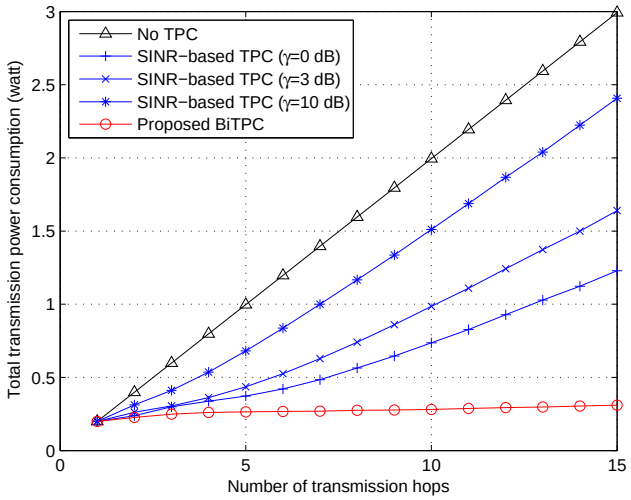


Figure 6. Total transmission power vs. number of hops.

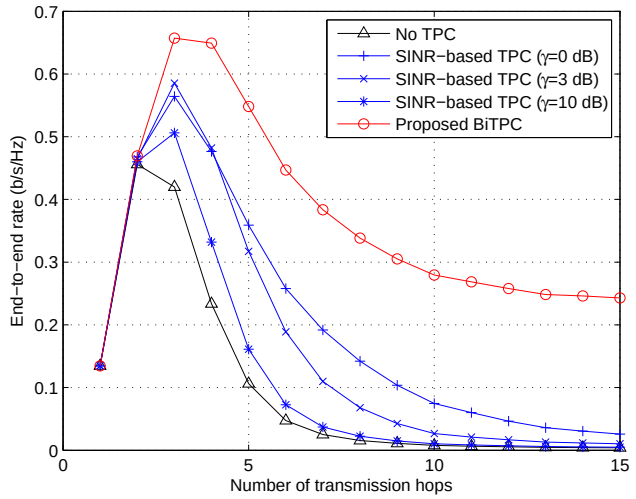


Figure 5. End-to-end rate vs. number of hops.

the proposed TPC algorithm dynamically achieves a SINR value that maximizes the end-to-end throughput, while the SINR-based TPC algorithm achieves a static target SINR. As the number of hops increases, the end-to-end rate increases sharply, but eventually decreases and maintains a constant level in both schemes. The increase in the number of hops initially improves the link budget and thus enhances the end-to-end rate, but the excessive number of hops causes more interference and degrades the end-to-end rate. This implies that not only the optimal TPC but also the optimal selection of transmission hops is required for maximizing the end-to-end rate in the given multi-hop environment.

Fig. 6 shows the performance of total transmission power consumption of all the transmitting nodes (i.e., the sum of the transmission power of each node) versus the number of transmission hops. The scheme without TPC uses a fixed maximum transmit power in all the nodes, so the total power consumption increases linearly according to the number of hops. On the other hand, the SINR-based and the proposed TPC algorithms use decreased transmission power. Particularly, the proposed BiTPC reduces the transmit power adaptively depending on the increase of interference due to the increased number of hops, and therefore, it exhibits very low power consumption. Note that the SINR-based TPC shows a tradeoff in performance between the end-to-end throughput and the total transmission power consumption according to the target SINR value.

VII. Conclusion

Inspired by the flocking algorithm, we proposed the BiTPC algorithm, which determines the transmit powers of nodes that equalize all the link rates on the multi-hop path. We proved that this rate-averaging algorithm is an optimal solution to maximize the multi-hop end-to-end rate in wireless networks with the solidarity property. The simulation results showed that the proposed BiTPC algorithm has the converged performances, regardless of the number of transmission hops and the information-sharing range, and leads to significant energy savings at the transmitting nodes by adjusting the transmit powers. Since the BiTPC algorithm is basically simple, distributed, and optimal, we expect that it will be practically used in complex and unstructured network environments. For further study, we will extend the basic concept of our BiTPC algorithm in the linear topology to the environment where multi-flow exists in the two-dimensional multi-hop topology.

Acknowledgment

This work was supported by the GRRC program of Gyeonggi province [(GRRCHankyong2011-B03), Low Power Machine-to-Machine Communication and Network for Management of Logistic Center], by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0025424), and by the Human Resources Development program(No.20124030200060) of the Korea Institute of Energy Technology Evaluation and Planning(KETEP) grant

 ISBN: 978-1-61208-318-6

funded by the Korea government Ministry of Trade, Industry and Energy.

REFERENCES

[1] T. ElBatt and A. Ephremides, “Joint Scheduling and Power Control for Wireless Ad Hoc Networks,” IEEE Trans. on Wireless Communications, vol. 3, no. 1, Jan. 2004, pp. 74-85.

[2] E.-S. Jung and N. H. Vaidya, “Power Control in Multi-Hop Wireless Networks,” Technical Report, Mar. 2002.

[3] P. Karn, MACA - a new channel access method for packet radio, 9th ARRL Computer Networking Conference, 1990, pp. 134-140.

[4] M. B. Pursley, H. B. Russell, and J. S. Wysocarski, “Energy-efficient transmission and routing protocols for wireless multiple-hop networks and spread-spectrum radios,” EUROCOMM, 2000, pp. 1-5.

[5] S. Agarwal, S. Krishnamurthy, R. H. Katz, and S. K. Dao, “Distributed power control in ad-hoc wireless networks,” IEEE PIMRC, vol. 2, Sep. 2001, pp. F-59-F-66.

[6] J.-P. Ebert, B. Stremmel, E. Wiederhold, and A. Wolisz, “An energy-efficient power control approach for WLANs,” Journal of Communications and Networks (JCN), vol.2, no.3, Sep. 2000, pp. 197-206.

[7] J.-P. Ebert and A. Wolisz, “Combined tuning of RF power and medium access control for WLANs,” IEEE Int. Workshop on Mobile Multimedia Communications, Nov. 1999, pp. 74-82.

[8] P. Lettieri and M. B. Srivastava, “Adaptive frame length control for improving wireless link throughput, range, and energy efficiency,” IEEE INFOCOM, vol. 2, Mar. 1998, pp. 564-571.

[9] N. Poojary, S. V. Krishnamurthy, and S. Dao, “Medium access control in a network of ad hoc mobile nodes with heterogeneous power capabilities,” IEEE Int. Conf. on Communications, vol. 3, June 2001, pp. 872-877.

[10] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, “Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol,” European Wireless 2002, Feb. 2002, pp. 1-7.

[11] A. Behzad and I. Rubin, “High Transmission Power Increases the Capacity of Ad Hoc Wireless Networks,” IEEE Trans. on Wireless Commun., vol.5, no.1, Jan. 2006, pp. 156-165.

[12] D. Choi and D. Park, “Effective self interference cancellation in full duplex relay systems,” IET Electronics Letters, vol. 48, no. 2, Jan. 2012, pp. 129-130.

[13] F. Cucker and S. Smale, “Emergent Behavior in Flocks,” IEEE Trans. on Automatic Control, vol. 52, no. 5, May 2007, pp. 852-862.

[14] J. Park, H. J. Kim, and S.-Y. Ha, “Cucker-Smale Flocking With Inter-Particle Bonding Forces,” IEEE Trans. on Automatic Control, vol. 55, no. 11, Nov. 2010, pp. 2617-2623.

[15] H. P. Thien, M. A. Moelyadi, and H. Muhammad, “Effects of Leader’s Position and Shape on Aerodynamic Performances of V Flight Formation,” ICIUS 2007, Bali Indonesia, Oct. 2007, pp. 43-49.

[16] C. J. Cutts and J. R. Speakman, “Energy savings in formation flight of Pink-footed Geese,” Journal of Experimental Biology, vol. 189, no. 1, 1994, pp. 251-261.

[17] H.-C. Le, H. Guyennet, and V. Felea, “OBMAC: An Overhearing Based MAC Protocol for Wireless Sensor Networks,” Int. Conf. on SensorComm, Oct. 2007, pp. 547-553.

[18] B. Radunovic and J.-Y Le Boudec, “Rate Performance Objectives of Multihop Wireless Networks,” IEEE Trans. on Mobile Computing, vol.3, no.4, Oct. 2004, pp. 334-349.

[19] 3GPP, “Technical specification group radio access network; further advancements for E-UTRA physical layer aspects (release 9),” TR 36.814 v9.0.0, Mar. 2010.

 ISBN: 978-1-61208-318-6

Tree Structured Group ID-Based Routing Method for Mobile Ad Hoc Networks

Hiroaki Yagi, Eitaro Kohno, and Yoshiaki Kakuda
Graduate School of Information Sciences, Hiroshima City University
Email: {yagi@nsw.info., kouno@, kakuda@}hiroshima-cu.ac.jp

***Abstract*—In ad hoc networks, wireless links can be disconnected by both the wireless instability and the node mobility. To tackle the wireless instability problem, multipath routing methods have been proposed. The multipath source initiated tree-based routing ID routing method (SRIDR) employs a tree-structured node ID that is assigned from a specific source node. While SRIDR is effective in compensating for disruptions due to wireless instability, it is not effective when the nodes are mobile. To counter disruptions due to mobile nodes, nodes have to reconstruct paths. Tree structured IDs can be assigned by each node in a self-organized manner. In this paper, we proposed a new tree-structured group ID-based routing method for mobile ad hoc networks. We have confirmed that our proposed method can maintain a high data delivery ratio even if the nodes are mobile.**

Keywords-Tree structured group ID; Mobile ad hoc networks; Bottom-up ID re-assignment.

I. INTRODUCTION

Ad hoc networks are self-organized networks, which consist of wireless terminals with routing and forwarding functions. In ad hoc networks, links between nodes can be disconnected due to both the instability of wireless connections and the node's mobility. These disruptions decrease the data delivery ratio. In previous research, we proposed the first source node initiated tree structured ID-based multipath routing method, referred to as the source initiated tree-based routing ID routing (SRIDR) [1] [2] in this paper. While SRIDR is an effective method to compensate for the instability of wireless connections, it is poor at compensating for disconnections due to node mobility.

SRIDR's ID assignment process employs the dynamic address routing (DART) [3] process. While this process is useful for static ad hoc networks, this process is problematic for mobile ad hoc networks. In mobile ad hoc networks, SRIDR's ID assignment process lead to the reassignment of all nodes in a network.

Jain et al. [4] proposed for assigning node ID in heterogeneous systems of wired access networks with mobile terminals. It provides a reassignment system for mobile terminals' IDs. It utilizes the same ID assignment process as the one described in [5]. [4] also utilizes the more effective "bottom up" method presented in [6] [7].

In this paper, we propose a new group ID-based multipath routing method that is an extension of SRIDR. Our proposed method employs group-based ID and "bottom-up" ID reassignment processes to improve the data delivery ratio of (mobile) ad hoc networks. We implemented our proposed method on a network simulator and confirmed the effectiveness of our proposed method.

The rest of the paper is organized as follows: in Section II, we discuss the background of our research. In Section III, we describe our proposed method. In Section IV, we show the results of our simulation experiments. We conclude the paper in Section V.

II. BACKGROUND

SRIDR is a multipath routing method for establishing multiple detouring paths that keeps a communication path connected even if some nodes fail in their data packet forwarding. SRIDR employs tree-structured node IDs in order to suppress the control packet count. A node ID is a unique identifier that shows the relationship between two or more nodes. Each node is assigned its node ID from the first source node [1] [2] [3], named the start node. In this paper, this assignment procedure is referred to as the "top down ID assignment procedure." In SRIDR, a node constructs its routing table using node IDs or subnet IDs. A node forwards data packets using the constructed routing table. A subnet ID shows a group of nodes which share the same node ID prefix. SRIDR has the following characteristics:

- Top down assignment of node IDs
- The utilization of routing tables to construct detour paths for data packets

In SRIDR, a node ID is assigned by a top down assignment process from the most significant bit of the node ID. Fig. 1 shows a network with nodes that have assigned node IDs. In SRIDR, the first source node (S) becomes the start node, which initiates the ID assignment process. The start node has the node ID (000) and it assigns a new node ID to adjacent nodes. Node C is an adjacent node of node S in Fig. 1. As an example, let us imagine node C is the first node in the network to make an ID request. In this case, node S assigns the node ID (100) to node C. Then node A, also an adjacent node of node S makes an ID request. So node S assigns the node ID (010) to node A. When a node is assigned its node ID, it can assign node IDs to adjacent nodes. Subsequently, in Fig. 1, node A assigns node B the node ID (011) and node C assigns node D the node ID (110). This is how nodes are assigned their node IDs. Nodes with assigned IDs exchange routing entries and make a routing table that constructs multiple paths. A node utilizes this routing table to find detouring paths for data packets. Thus, SRIDR can compensate for wireless instability by using detouring paths. In mobile ad hoc networks, node ID based detouring paths are

 ISBN: 978-1-61208-318-6

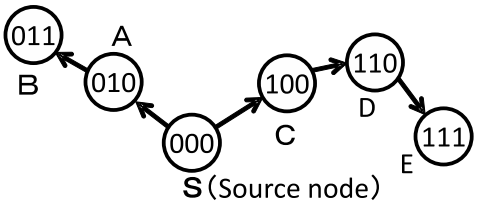


Figure 1: "Top-down" node ID assignment procedure.

usually invalid for SRIDR because node IDs might differ in more than one bit and therefore cannot connect and forward data. To counter this, node IDs must be reassigned. However, if every node is reassigned, the resulting influx of control packets will reduce the data packet delivery ratio. In addition, when a reassignment request occurs too frequently on a specific node, the node cannot have a stable node ID and therefore cannot forward data appropriately. For this reason, we have proposed the following method.

III. Proposed Method

A. Overview

In this paper, we propose a new group ID assignment method in order to deal with the reduction of the data delivery ratio due to node mobility in the tree-structured group ID. In our proposed method, nodes share the same ID and constructs groups to minimize the frequency of assignment requests. We propose reassigning group IDs in a self-organized manner, referred to as the "bottom-up" reassignment process. The "bottom-up" reassignment process is localized to the parent-child relationship of the mobile nodes. This reassignment process can suppress the escalation of control packets. Our proposed method has the following characteristics:

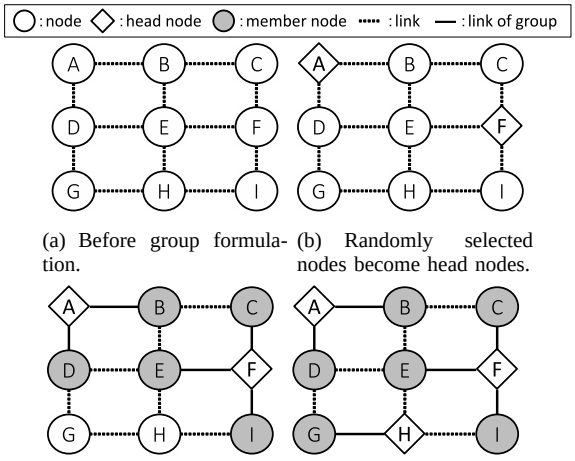
1) Group formulation
2) "Bottom-up" group ID assignment
3) Group ID reassignment

In our proposed method, nodes first form a group. After that, each group is assigned its initial group ID. Adjacent groups exchange information to form a tree structure of group IDs. When the links between two or more groups are disconnected, our method reassigns group IDs. We describe our proposed method below.

B. Group formulation

In our proposed method, multiple nodes that have the same node ID form a group. Nodes in a group are categorized as either a head node or member nodes. The formulation of groups is as follows:

1) Randomly selected nodes in the network become head nodes.
2) The adjacent nodes to the head nodes become member nodes.
3) If a node is not a head or a member node, it performs the above-mentioned procedures (1 and 2) until all nodes become either head or member nodes.



(a) Before group formulation.
(b) Randomly selected nodes become head nodes.
(c) The adjacent nodes to head nodes become member nodes.
(d) Group formulation completed.

Figure 2: Group formulation procedure.

Fig. 2 shows the group formulation procedure. Fig. 2(a) shows an example topology of a network. In Fig. 2(b), nodes A and F are randomly selected to become head nodes. Subsequently, the nodes adjacent to A and F become member nodes. This is shown in Fig. 2(c). The solid black line denotes this relationship. In Fig. 2(c), nodes B and D become member nodes of node A. Node C, E, and I become member nodes of node F. These procedures are repeated and the network forms groups such as Fig. 2(d).

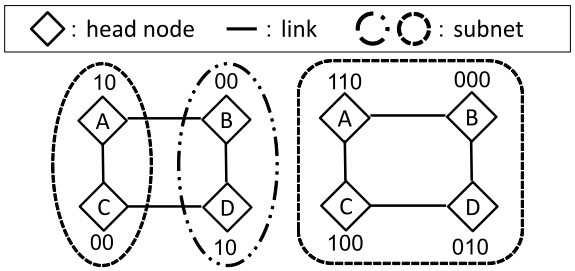
C. "Bottom-up" group ID assignment

When the network forms groups, the head nodes start group ID assignment procedures. In contrast with the "top down" ID assignment process, the "bottom-up" ID assignment process assigns group IDs from the least significant bit. When the head node in a group is assigned its group ID, the member nodes share the same group ID as the head node. In the procedure of assigning group IDs between two head nodes, one head node establishes itself as a parent head node and the other as a child head node. At this time, parent and child head nodes and their member nodes form a subnet. A subnet is a small group of nodes which shares the same prefix as the larger group ID. After that, each group starts to construct a group ID tree using the following procedures:

1) Every head node has the initial group ID, (0).
2) Each head node searches its adjacent head node to construct a subnet.
3) The constructed subnet searches for its adjacent head nodes or subnets to construct a new larger subnet.
4) Two subnets/head nodes construct a new larger subnet by adding 0 or 1 at the beginning of the assigned group ID.
5) Procedures 2-4 are performed repeatedly during the predetermined time.

Fig. 3 shows the procedures of group ID assignment. For convenience, Fig. 3 shows only head nodes. Each head node has an initial group ID, (0), and constructs a new subnet by combining with other head nodes. Suppose that head nodes A

 ISBN: 978-1-61208-318-6



(a) Conbine between head nodes. (b) Conbine between subnets.

Figure 3: Group ID assignment procedure.

and C combine. If , for example, node C is the first to request combining, node A becomes the parent head node, and assigns a 1 in front of its assigned group ID, (0). Then node C becomes the child head node and it assigns a 0 in front of the assigned group ID, (0). As a result, the group IDs of nodes A and C become (10) and (00), respectively. In the same manner, head nodes B and D form a new subnet and become child and parent head nodes as shown in Fig. 3(a).

Next, we assume that subnets A-C and B-D will combine and construct a new subnet. When head node A becomes the parent head node, head nodes in subnet A-C add a 1 at the beginning of their existing group IDs (10) and (00). In addition, head nodes in subnet B-D add a 0 at the beginning of their existing group IDs (00) and (10). Consequently, head nodes A, B, C, and D acquire their group IDs (110), (000), (100), and (010), respectively.

D. Group ID reassignment

When a child head node or its member node detects a disruption in the link to its parent head node, the child head node or member node starts the group ID reassignment procedure. While the parent head node also detects a disruption in the link with the child head node or member node, it does not perform the group ID reassignment procedure, maintaining a higher packet delivery ratio. The following is the group ID reassignment procedure in our proposed method:

1) A child head node or a member node detects a disruption in the link to its parent head node.
2) If a child head node or a member node locates another head node, it reassigns its group ID to join the new head node. This completes the group ID reassignment procedure.
3) If a child head node or a member node cannot locate another head node, it starts the group ID reassignment procedure with the member nodes in its group.
4) After procedure 3, the "bottom-up" group ID assignment procedure will start and the child head node's group will join the group ID tree of an adjacent group.

Fig. 4 shows the result of this group ID assignment procedure. In Fig. 4, X indicates a "don't care bit" that is a 1 or a 0. "1XX" shows the subnet that includes head nodes A and C, and member nodes E and G. These nodes shares the same prefix "1" at the beginning of their group ID/node IDs. Similarly, "0XX" shows the subnet that includes head nodes B and D, and member nodes F and H. These nodes share the same prefix "0" at the beginning of their group ID/node IDs. "XXX" indicates the whole network. It is special a subnet.

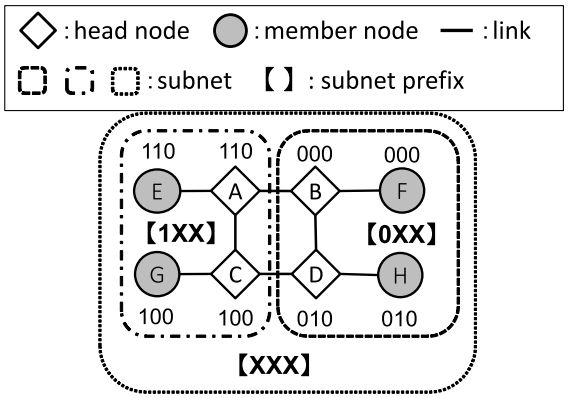


Figure 4: Result of group ID assignment procedure.

E. Data packet forwarding

As in SRIDR, our proposed method forwards data packets in the following way:

1) A node compares its group ID with the destination node's group ID, and calculates the subnet to which the destination node belongs.
2) The node refers its own routing table and determines the next hop node using the subnet information.
3) The node forwards the data packet to the next hop node.
4) Our proposed method repeats procedures 1-3 until the destination node receives the data packet.

IV. Simulation Experiments

A. Overview

We have conducted simulation experiments of our proposed method on a network simulator, QualNet ver.5.0 [8], to evaluate its effectiveness. We have measured the data packet delivery ratio and the control packet count. We compared our proposed method to SRIDR and a tree structured bottom-up node ID-based routing method without group IDs.

B. Simulation Environment

TABLE I: Parameters for experiments.

Simulator	QualNet ver.5.0 [8]
Number of nodes	50
Field size (m^2)	1100 × 1100
Number of source and destination pairs	1, 5
Data packet size (Byte)	512
Data packet interval (sec)	0.25
Radio area (m)	250
MAC layer protocol	IEEE802.11b
Max. bandwidth (Mbps)	2
Simulation time (sec)	2000
Node distribution	RANDOM
Max. node speed (m/s)	5
Node mobility model	Random waypoint model [9]

Table I shows the parameters for the simulation experiments. We set the field size to accommodate an average of

 ISBN: 978-1-61208-318-6

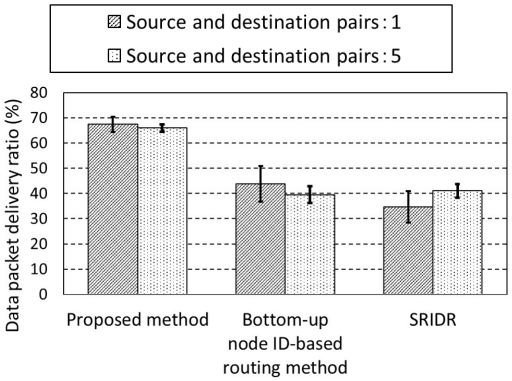


Figure 5: The data packet delivery ratio.

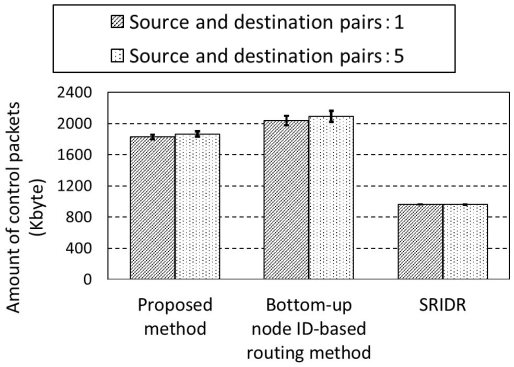


Figure 6: The amount of control packets.

eight adjacent nodes. In addition, nodes are randomly deployed and there is one source and destination pair (In Figs. 5 and 6 we denoted as “Source and destination pairs:1 ”) or five source and destination pairs (In Figs. 5 and 6 we denoted as “Source and destination pairs:5 ”).

C. Methodology

When a run of the simulation experiment starts, each node starts group formulation procedures and performs bottom-up group ID assignment procedures. 20 seconds after the start of the simulation run, the first source node starts transmitting the data packets. With the configuration of five source and destination pairs, 10 seconds after that, the second source node starts transmitting data packets. Thereafter the other source nodes start to transmit data packets one after another every 10 seconds. When a node detects a disruption, SRIDR sends data packets using detouring paths. In contrast, our proposed method and the tree structured bottom-up node ID-based routing method performs their group ID/node ID reassignment processes instead.

D. Result

Figs. 5 and 6 show the results of the data packet delivery ratio and the control packet count with respect to our proposed method, SRIDR, and the tree structured bottom-up node ID-based routing method, respectively. The vertical axes show the results of the data packet delivery ratio and the control packet count, respectively. The horizontal axes are the routing methods. Each result is the average of 50 simulation runs. The error bars show 95 percent confidence intervals.

To measure the amount of control packets, we created an output file in a simulator. In the simulator, when a control packet was transmitted, the simulator added a line to the output file indicating the data size of the control packet. When the simulation experiment was concluded, we calculated the total amount of control packets.

Control packets varied in size from 8 to 28 (Byte/packet). However, control packet containing node routing tables varied with a larger range.

E. Discussion

Fig. 5 shows our proposed method has the highest data packet delivery ratio among the methods. This shows that our proposed method is the most effective. Additionally, the data packet delivery ratio of SRIDR and that of the tree structured bottom-up node ID-based routing method are almost the same. In the tree structured bottom-up node ID-based routing method, there is a high frequency of reassignment procedures which slows down data packet delivery ratio to about the same as SRIDR. Therefore, the data packet delivery ratio is the same as SRIDR.

Fig. 6 shows that the control packet count of SRIDR is the smallest and the control packet count of our proposed method is slightly smaller than that of the tree structured bottom-up node ID-based routing method. In SRIDR, since the node ID reassignment procedures were not performed, the control packet count of SRIDR is smaller than that of our proposed method. While the control packet count of our proposed method was larger than SRIDR, the frequency of group ID reassignment was low. In our proposed method, links between groups were increased by forming groups, leads to a smaller control packet count than that of the tree structured bottom-up node ID-based routing method.

V. Conclusion and Future Work

In this paper, we proposed a new tree structured group ID-based routing method to tackle the disruptions caused by the node mobility. We implemented our proposed method on a network simulator and performed the experiment to compare and evaluate the results. Our proposed method showed a high data delivery ratio and lower control packet count than that of the tree structured bottom-up node ID-based routing method.

As future work, we planned to measure characteristics over time of our proposed method. We have to improve our method taking into account the connectivity of groups.

Acknowledgment

This research is supported by JSPS KAKENHI Grant Number (B) (No.24300028), and (C) (No.25330109). This research is also supported by the Ministry of Internal Affairs and Communications under Strategic Information and Communications R&D Promotion Programme (SCOPE) Grant No.131408006, and by The Telecommunications Advancement Foundation (TAF), Japan.

 ISBN: 978-1-61208-318-6

REFERENCES

[1] T. Okazaki, E. Kohno, T. Ohta, and Y. Kakuda, “A multipath routing method with dynamic ID for reduction of routing load in ad hoc networks.” in ADHOCNETS’10, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, J. Zheng, D. Simplot-Ryl, and V. C. M. Leung, Eds., vol. 49, no. 2, 2010, pp. 114–129.

[2] T. Okazaki, E. Kohno, and Y. Kakuda, “Improvement of assurance for wireless sensor networks using packet detouring and dispersed data transmission,” Proceedings of the 2011 IEEE International Conference on Internet of Things and Cyber, Physical and Social Computing (iThings/CPSCom 2011), 2011, pp. 144–151.

[3] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, “DART: Dynamic address routing for scalable ad hoc and mesh networks,” IEEE/ACM Transactions on Networking, vol. 15, no. 1, Apr. 2007, pp. 119–132.

[4] S. Jain, Y. Chen, and Z.-L. Zhang, “Viro: A scalable, robust and namespace independent virtual id routing for future networks,” in INFOCOM, 2011 Proceedings IEEE, 2011, pp. 2381–2389.

[5] L. Ramachandran, M. Kapoor, A. Sarkar, and A. Aggarwal, “Clustering algorithms for wireless ad hoc networks,” in Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, ser. DIALM ’00. New York, NY, USA: ACM, 2000, pp. 54–63.

[6] S. Jain, Y. Chen, Z.-L. Zhang, and S. Jain, “Veil: A ”plug-&-play” virtual (ethernet) id layer for below ip networking,” in GLOBECOM Workshops, 2009 IEEE, 2009, pp. 1–6.

[7] G.-H. Lu, S. Jain, S. Chen, and Z.-L. Zhang, “Virtual id routing: A scalable routing framework with support for mobility and routing efficiency,” in Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture, ser. MobiArch ’08. New York, NY, USA: ACM, 2008, pp. 79–84.

[8] Scalable Network Technologies, Inc., “QualNet network simulator.” [Online]. Available: http://www.scalable-networks.com/ [retrieved: December, 2013]

[9] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in Mobile Computing. Kluwer Academic Publishers, 1996, vol. 353, pp. 153–181.

 ISBN: 978-1-61208-318-6

Routing Algorithm for Automatic Metering of Waterworks Data

Gang-Wook Shin Ho-Hyun Lee Sung-Taek Hong Jae-Rheen Yang
K-water Research Institute
K-water(Korea Water Resources Corporation)
Daejeon, South Korea
gwshin@kwater.or.kr lhh@kwater.or.kr sthong@kwater.or.kr jyang@kwater.or.kr

Abstract—**Real-time acquisition of water meter data is very difficult owing to the expensive and difficult environment for installing sensors and wired or wireless communication systems. In this study, a wireless network algorithm that takes into consideration technical and economic factors is proposed. Cluster-based ZigBee communication method is applied to transmit the sensing data without disruption regardless of environmental situations. In addition, it is designed to select the cluster head considering the minimization of battery power consumption. Through this study, data pertaining to the characteristics and the economical efficiency of water meters can be collected and analyzed. This study shows that the system can be widely used to enhance the water distribution networks for stable water supply system.**

Keywords- formatting; water data, automatic metering, meter reading system, block system

I. INTRODUCTION

The data collected from water meters are used basically to ensure stable water supply. The data is very important because it is used to determine the tap water fare for individual consumers. However, until now, inspectors have had to directly visit meters rather than real-time data acquisition because of cheap water fare. The irregularity of data collection could result in inefficient manufacture and supply of water and equitable water allocation might not work in a country which experiences water shortages. In addition, many complaints are associated with water supply charges that are based on irregular data.

With the development of information and communication technologies, the reliable information to the general public is provided by realization of a ubiquitous environment in the field of electricity, gas, and water utilities. In general, providing accurate and up-to-date information pertaining to water usage is very difficult and costly.

In Korea, most of the water meters have been installed in poor environments, such as basements or underground areas that are hard to access. Also, most of the water meters are of the inexpensive mechanical type. Mechanical water meters can only be read by visiting the household where it is located. In addition, because the meters do not support remote automatic meter transmission, there are various problems associated with individual visits, such as errors relating to minimum flow, consumer privacy, recording errors, and re-visits due consumer absence.

In Europe, most of the water meters are installed in easily accessible spaces. However, as opposed to Korea, a water meter can stably transmit real-time data if a ubiquitous sensor network is used. But mechanical water meter types are being used due to economic factors in Europe.

The real-time remote automatic meter reading system, i.e., Automatic Meter Reading (AMR), has so far been unable to be introduced in Korea due to poor installation sites and water cost [1-3].

Networks for real-time transmission of data can be designed using telephone networks, Cable Television (CATV) networks, Code Division Multiple Access (CDMA) networks, Radio Frequency (RF) networks, and Power Line Communication (PLC) networks, but the Industrial Scientific and Medical (ISM) band RF communication network for short distances and the CDMA for long distances have been proposed considering cost and reliability. RF repeaters for short-range communication are required depending on the propagation environment, but there are challenges such as installation conditions and power management.

Therefore, the cluster-based routing algorithm is proposed to enable real-time data transfer considering the power consumption in this study. The status of the water meter data transmission is explained in Section 2 of this study, and the proposed routing algorithm is described in Section 3. The performance of the proposed method is measured and analyzed through NS-2 simulation in Section 4 and finally, Section 5 comprehensively analyzes the feasibility and applicability of the algorithm.

II. DATA TRANSMISSION SYSTEM FOR WATER METERS

There are a variety of data collection methods that are used in the field. Meter reading can be done through direct readings taken during home visits. In outdoor readings, the same data are sent to an external display device from indoor water meters. Meter inspectors obtain data with the help of wireless communication device by walking-by or driving-by the properties. Water meters can be automatically read from a fixed communication network. A typical real-time data transmission system used by public utilities, such as electric, gas, and water, consists of the sensor unit, the transmission unit and data management part as shown in Figure 1. The function of each component is as follows. At first, the sensor unit measures the object and converts the mechanical data into electrical data that can be delivered. Then, they are sent to a transmission unit to transfer them to data management systems, which store the transmitted data simultaneously.

 ISBN: 978-1-61208-318-6

The transmission unit that collects data from a large number of sensors has the ability to send certain data on a regular basis and to manage their storage. All the data are transferred into a database and used for performing fare management, finding abnormal measurements, data analysis and supply forecasting for management systems.

The meter measures water usage as the key component in the sensor unit. The meter of each home has a diameter of 50 mm or less. Measurement methods could be classified into direct and indirect method. The indirect method depends on the rotation speed of the actual flow. The direct method, which is used primarily for testing, measures a constant volume of water. In order to transmit real-time data, a digital meter having a microprocessor and a memory is required.

Although a partial digital meter has a mechanical type of sensor, the meter can convert mechanical signals into electrical signals. However, the meter does not have an internal microprocessor for data processing. There are two kinds of partial digital meters. One is a pulse counting method that uses a lead switch, and the other is the camera method which has the ability to take and send photos for image processing. The camera method should be equipped with a microprocessor and a memory to transmit real-time data.

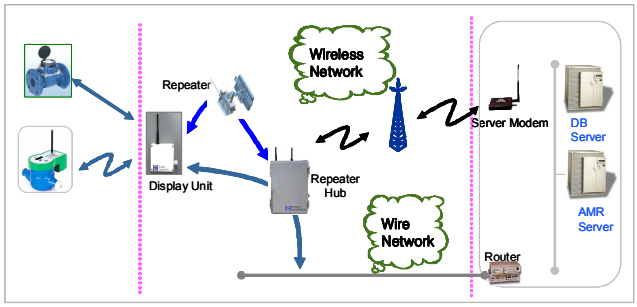


Figure 1. Schematics of Water Data Transmission

Data transmission methods can be divided into the wired and the wireless method. The latter has the advantage of affordability in terms of installation, equipment and communication circuit, but it has the disadvantage that data is unreliable because of communication interference. So, the wireless communication and the power line communication are being widely adopted using ubiquitous environment. As such, communication environments and meter reading methods should be considered not to lose water meter data.

III. CLUSTER-BASED ROUTING ALGORITHM

Pattern of a residential area shown in Figure 2 must be considered to make a plan for sensor distribution. Because the operation units for water supply and management are generally divided into blocks, the block can be divided depending on block's size. The shape of the block can be split into squares, rectangles, triangles, polygons, etc. In this study, a suitable transmission algorithm for a variety of patterns is proposed.

An environment for the installation of repeaters cannot always be located in the middle of the block for obtaining meter data remotely. They are located in various locations, such as center, corners or edges under the terms of the block. Thus, the nodes are not always generated with configuration of a star network. It is not economical to install each repeater for these nodes, since it will be an obstacle for optimization of transmission environment. Therefore, it is more efficient to transfer after collecting data through cluster formation.

Figure 2. Pattern of a Residential Area

The nodes can be divided into a small area of the sensor field called clusters. Each cluster has a cluster head which collects data from the cluster members and the data is delivered to the different clusters or the relay station [4-7].

In particular, the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol, which is most commonly used to maximize the survival time of the network, adopts the probability function as the following equation, which keeps energy consumption evenly between the network nodes.

$$P_i(t) = \begin{cases} \dfrac{k}{N - k^*\left(r \bmod \frac{N}{k}\right)} & : C_i(t) = 1 \\ 0 & : C_i(t) = 0 \end{cases} \quad (1)$$

Here, i is the identifier of the node, t is the time, N is the total number of nodes, k is the number of clusters and r represents the round. This protocol consists of advertisement, cluster setup, schedule generation and data transfer step. If the head is selected at least once during r mod (N / k) rounds in the above equation, the chance of re-selection is zero. In case that selected cluster head cannot be delivered to sink node via one-hop, it is difficult to expect a high transmission rate. In this study, the following cluster head selection method is proposed.

 ISBN: 978-1-61208-318-6

In order to determine the environmental diversity of remote water meter reading like the distribution pattern of a residential area, such as in Figure 2, the network should be initialized as shown in Table 1. First, the repeater broadcasts an advertisement message to characterize the entire node. Second, it gives the node IDs according to each node type, and third, it performs the function of a cluster head selection and change. The whole network becomes ready to start.

TABLE I. CHARACTERISTICS OF ADVERTISEMENT MESSAGES

Node_ID	Unique ID assigned to a node
Hops	Hop depth
Energy_Data	Residual energy of the node
SN_Limit	Signal level limits

Hops mean the hop depth from the repeater station, which is represented by the number of hops. The procedure to create the table about their neighbor nodes is done by knowing the depth of each node. Their repeater station sets zero for hop depth, and then advertisement message is transmitted to the network. The node to receive an advertisement message from the repeater sets its own hop to 1, and then it locally broadcasts an advisement message including hop information, residual energy, and Node_ID. If the node is not yet able to secure the depth of the hop among received nodes, it adds one to the value of the received hop set. If the node knowing the depth of hop receives an advertisement message, it stores the nodes less than the depth of its hop information in the parent node table, such as in Table 2. If the hop depth of the received message is the same, this information should be stored in the neighbor node table.

The parent node table is used to transmit data from the cluster node to the cluster head on the top. If there is no parent node table with their lower-hop value than their own table, it can be transmitted by searching for a detour route to refer to the neighbor node table with the same-hop value.

TABLE II. NODE TABLES

Table name	Relationship between value of hops
Parent node table	My hop < Received nod hop
Neighbor node table	My hop = Received node hop
Child node table	My hop > Received node hop

After the child node table has stored the paths of higher nodes than their own hops, it refers to the reverse path to send messages such as advertisements, broadcast and a control messages between random nodes.

In addition, it is transmitted by specifying a range of signal levels in the advertisement message to distinguish the type of node due to the level of the signal repeater. The configuration of the network is desirable to have the composition of a star and cluster to obtain real-time data. Thus, the node and repeater with a star topology can transmit data directly without the other nodes. However, nodes outside the range of the signal level of any node in the cluster are composed of local-based clusters to transmit data to repeaters through a cluster head. Repeater gives the information of node types based on the signal levels of all nodes to distinguish the specifications of such transmission methods. The node type is specified in the repeater signal level and has the attribute value of Table 3.

TABLE III. CHARACTERISTIC VALUES OF NODES

Size of signal level	> -50 dBm	< -50 dBm
Attribute value of node type	Inside	Outside

Nodes with inside attribute values collect node data with outside attributes and they play a role as a cluster head to send to repeaters or send their own data directly to repeaters. Outside nodes will form a local based cluster to receive and transmit data to nodes selected as the cluster head in the transmission path.

Among the nodes with inside property, any node with a large signal level intensity and within one hop to a repeater is chosen as a cluster head. Locally distributed 2 to 3 cluster heads are selected considering the number of nodes and positions, which are then registered as cluster head members in the node.

The nodes registered in the cluster head members will have the property value of activity and standby cluster headers, such as in Table 4. An active cluster head will be selected by calculating the probability among cluster members in each round. The node working as the active cluster head aggregates data in a cluster to be sent to the repeater, and the rest of the cluster members have the property value of a standby cluster head to send only their own data to the repeater.

TABLE IV. CONFIGURATION OF CLUSTER MEMBER

Activity cluster head	Transmission of aggregated data in the cluster
Standby cluster head	Transmission of their own data

Any number of the nodes with the outside attribute value become one cluster considering geographic location with a cluster member, such as in Table 5.

The cluster heads broadcast their own state to all nodes in the scope of the area. The node to receive the broadcasting message checks which node is selected as the cluster head among cluster members. Then the node prepares for sending data to the selected head.

Each cluster member node is required to have a routing table to send a message to the sensor node. Repeater node, which sends data messages to the cluster head, stores lower

 ISBN: 978-1-61208-318-6

node information in the child node table. It can be used as routing information when sending data to the header. At the same time, repeaters replace the value of their own ID and hop depth with the Destination_ID and Destination_Hops of the data message before sending them to the heads.

TABLE V. CLUSTER MEMBERS

Cluster member	Cluster division	Nodes configuration in the cluster
CMember1	1# cluster	Node0, Node1, ..., Node24
CMember2	2# cluster	Node25, Node26, ..., Node49
CMember3	3# cluster	Node50, Node51, ..., Node74
CMember4	4# cluster	Node75, Node76, ..., Node99

Data refer to the parent node and neighboring node table to do flooding in the direction of the lower node in hop depth. If a routing node receives a data message, it first compares its depth and Destination_Hops, and if the hop depth is smaller than its depth, it transmits the data in the direction of the cluster head. The cluster head finally obtains the data through these processes.

TABLE VI. TRANSMISSION DATA FIELD

Destination_ID	Address of destination node.
Destination_Hops	Hop of destination node
DATA	Sensing data
Energy_Data	Amount of residual node energy

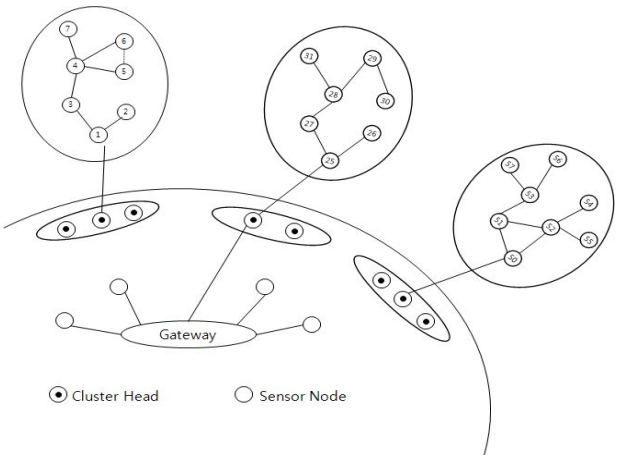


Figure 3. Routing topology

As shown in Figure 3, the topology is proposed for optimal routing algorithm. The algorithm selects a node with the best amount of residual energy.

IV. SIMULATION AND ANALYSIS

The network simulator, NS-2, was used to validate the proposed algorithm in this study [8]. 100 sensor nodes were randomly distributed in the area of 100m x 100m. The sink node was placed in the center of simulation area and the maximum transmission distance of the nodes was assumed to be 10m. In addition, the average packet transmission rate was set to 98% to consider the error rate of the local communication. Sensor nodes, except the sink node, were set to transmit the data packets during 2,030 seconds. The first 30 seconds is allotted to select cluster creation and cluster head and then the data packet is transmitted for the experiments five times at intervals of 400 seconds. The 100 sensor nodes were divided into four groups - 25 sensor nodes per each group. The sensor node that can transfer data to sink node by one hop was selected as the cluster head. To monitor the energy variances of each cluster head, the cluster head was set to change every 50 seconds.

Among Zigbee communications, the most widely used routing algorithm is the Ad-hoc On demand Distance Vector(AODV). The characteristic of this algorithm is the distance vector routing for each node to search the most optimal route from a source node to a destination node. Thus, this study compared the proposed algorithm with the AODV routing algorithm [9-12].

Data transmission interval of static type was set to 10 seconds. Those of multiple type were set to 1, 3, 5, 7 and 10 seconds for every 5 nodes in each cluster. Packet transmission rate is the ratio of the finally reached data packets to the generated data packets at the source node. Figure 4 shows that the packet transmission rate of the static type of 10-second cycle is higher than that of multiple type of various cycles. The proposed routing algorithm showed over 4% increased result in both static and multiple types comparing with AODV.

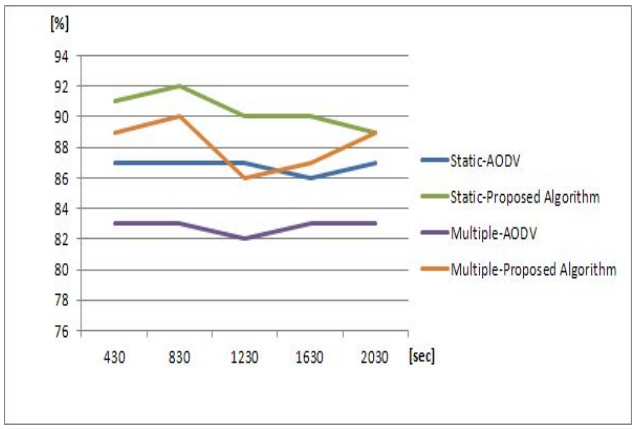


Figure 4. Packet transmission rate

In this simulator, the initial energy of each node was set to 20J. Each node transmits data to gateway through the cluster head at the transmission cycle. Transmit power and

 ISBN: 978-1-61208-318-6

received power were set to 0.0756J and 0.0828J, respectively, and the remaining energy was obtained after the transmission of 2,030 seconds. As shown in Figure 5, the nodes simulated by proposed algorithm had more remaining energy than those simulated by AODV algorithm by about 50%. The result shows that the proposed algorithm performs better in both static and multiple type cases.

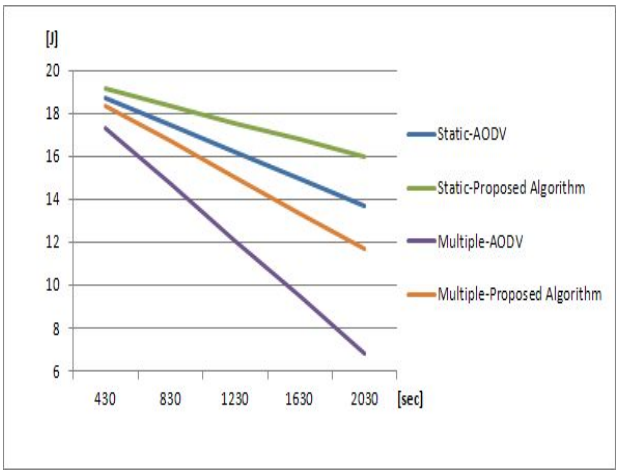


Figure 5. Minimum remaining energy

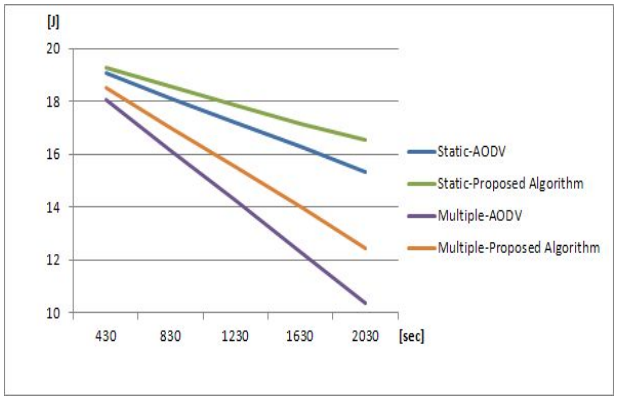


Figure 6. Average remaining energy

Figure 6 shows the average value of the remaining energy in all nodes after the simulation. The average remaining energy of proposed algorithm is about 20% more than that of AODV algorithm. We can notice that the static type has more remaining energy because the transmission cycle of static type is longer than that of multiple type.

V. CONCLUSION

This study proposed a cluster-based routing algorithm to obtain the real-time waterworks data. In particular, we considered the environment of communication and power with priority. Among the sensor nodes that transmitted to the sink node, each cluster's head was selected by checking the reference receiver signal level and remaining energy level.

The experimental results show about 4% improvements in packet transmission rate as compared to AODV algorithm. Because batteries are essential in waterworks data transmission environment, effective use of energy is very important. Compared to the existing AODV, the proposed method for effective use of energy shows improvements of more than 50% at maximum. The results of this study can be applied in realization of water reading system where real-time waterworks data transmission is possible.

In the future, the proposed algorithm may be applied to the implementation of the water management system through the expansion of water reading system.

REFERENCES

[1] F. Arregui, E. Cabrera Jr, and R. Cobacho, "Integrated Water Meter Management", IWA Publishing, 2006

[2] OIML, "International Recommendation R 49-1", 2003.

[3] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: a survey", In International Journal on Elsevier Computer Networks, Vol. 47, 2005, pp.445-487.

[4] G. W. Shin, S. T. Hong, and Y. W. Lee,"Walk-by Meter Reading System of Digital Water Meter Based on Ubiquitous", Journal of Control, Robotics and Systems Engineering, Vol. 15,2009,pp.668-693.

[5] M. U. Mahfuz and K. M. Ahmed, "A review of micro-nano-scale wireless sensor networks for environmental protection: Prospects and challenges", Science and Technology of Advanced Materials, Vol. 6 , 2005, pp.302-306.

[6] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, "Networking issues in wireless sensor networks", Journal of Parallel and Distributed Computing, Vol. 64, 2004, pp.799-814.

[7] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions on wireless communications, Vol. 1, No. 4, 2002, pp.660-670.

[8] The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/, Dec. 2013.

[9] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc On-demand Distance Vector(AODV) Routing", Internet-Draft, IETF, March, 2002.

[10] ZigBee Alliance, "ZigBee Specification: ZigBee Document 053474r17", 17 Jan. 2008.

[11] K. Akkaya and M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", In International Journal on Elsevier Ad Hoc Network, Vol. 33, 2005, pp.325-349.

[12] D.Y. Kim and W.S. Jung, "An Efficient Shortcut Path Algorithm using Depth in Zigbee Network, Vol. 34, 2009, pp.1475-1482.

 ISBN: 978-1-61208-318-6

MLSD: A Network Topology Discovery Protocol for Infrastructure Wireless Mesh Networks

Daniel Porto, Gledson Elias
Informatics Department
Federal University of Paraíba
João Pessoa, Brazil
d.porto@campus.fct.unl.pt , gledson@ci.ufpb.br

***Abstract*—In Infrastructure Wireless Mesh Networks (IWMN), the network topology discovery protocol has an essential role for responding proactively and promptly to topology modifications. It is responsible for disseminating link state updates, managing the tension between update frequency and number of messages, which has strong impact in protocol performance, network resource consumption and scalability. In such a context, this paper proposes the Mesh Network Link State Discovery (MLSD) protocol, which has been specifically designed considering IWMNs features. MLSD adopts a proactive, reliable, incremental, controlled and event-based delivery strategy, which avoids periodic messages and coordinates how information is propagated for enhancing efficiency. Besides, it joins several updates for reducing network resource consumption.**

Keywords-wireless mesh networks; routing protocols; link-state protocols

I. INTRODUCTION

As an evolution of wireless networks, *Infrastructure Wireless Mesh Networks* (IWMNs) have emerged as a key technology for dynamic self-configurable and self-healing networks that provide large-scale, reliable service coverage, allowing devices to automatically reconfiguring, establishing and maintaining connectivity among themselves [1][2]. In essence, an IWMN is a multi-hop wireless network that introduces a hierarchy of devices, called mesh routers and mesh clients [3]. *Mesh Routers* (MRs) are dedicated, stationary and power enabled devices, strategically positioned to provide a multi-hop wireless backbone for stationary or mobile, power constrained *Mesh Clients* (MCs).

In IWMNs, connectivity among non-neighboring nodes is achieved through multi-hop communication in which MRs forward packets hop by hop to other intermediate MRs in direction to the destination node. Note that, in IWMNs, MCs cannot forward packets and besides cannot communicate directly with each other. Thus, MRs have to manage and disseminate routing information, and to do that, a routing protocol must be adopted.

In practice, taking into account shared similarities among IWMNs and *Mobile Ad Hoc Networks* (MANETs), routing protocols designed for MANETs have been applied to IWMN projects [1]. For instance, the VMesh [4] project employs the *Optimized Link State Routing* (OLSR) protocol [5]. As another example, Microsoft mesh networks [6] are built using a modified version of the *Dynamic Source Routing* (DSR) protocol [7].

However, according to Akyildiz [1], ad hoc routing protocols do not scale very well in IWMNs and the throughput drops as the number of nodes increases. Therefore, despite the large availability of ad hoc routing protocols, considerable research efforts are still needed for designing more efficient and effective routing protocols for IWMNs, which can be specifically designed to explore and take advantage of their built-in features.

Taking into account IWMN architectural features, in order to be able to forward packets, MRs ought to support advanced routing capabilities, which need to detect fast topology changes and keep updated routes. In such a context, the link-state routing approach has the advantage of fast convergence when contrasted with the distance-vector routing approach, updating in a faster way routing information in all nodes of the network. Thus, in IWMNs, link-state routing protocols seem to be more adequate than distance-vector routing protocols.

In link-state routing protocols, a key element is the *Network Topology Discovery Protocol* (NTDP), in which network topology updates are propagated using messages called *Link-State Advertisements* (LSAs) [8]. Such protocols can generate LSAs on a periodic basis or can adopt an event-based approach for generating LSAs when detect changes in the state of the wireless links among nodes [9].

In such a context, this paper presents a scalable, robust and reliable network topology discovery protocol, called *Mesh Network Link State Discovery* (MLSD), based on the link-state approach and specifically designed taking into account IWMNs features. In order to reduce the control overhead related to topology update messages, the MLSD protocol adopts a proactive, reliable, incremental, controlled and event-based approach for generating LSAs, making more efficient use of network resources. Simulations show over 60% reduction in the control overhead of topology discovery compared to a periodic-based link-state protocol.

The remainder of this paper is organized as follows. Section 2 examines the strategies adopted in related work for disseminating topology information. Then, Section 3 presents the MLSD protocol, detailing its several strategies for disseminating link state updates. Next, Section 4 presents initial performance evaluation results, and, in conclusion, Section 5 draws final remarks and delineates future work.

 ISBN: 978-1-61208-318-6

II. RELATED WORK

In order to contextualize the proposed protocol, this section identifies routing protocols that have been adopted in IWMNs, highlighting their strategies for disseminating topological information. According to Chen [10], IWMNs represent a recent research field, favoring the adoption of ad hoc routing protocols, in special, the strategies employed by *Optimized Link State Routing* (OLSR) [5] and *Ad hoc On-Demand Distance Vector* (AODV) [11] protocols.

OLSR [5] is a proactive, link-state routing protocol that disseminates topological information using a flooding process that consists in choosing a set of *Multipoint Relays* (MPRs), which are responsible for periodically generating and forwarding topology control messages throughout the wireless network. Thus, OLSR simply floods topology data often enough to make sure that the topological database does not remain unsynchronized for extended periods of time.

In OLSR, MPRs are chosen considering the neighborhood of the nodes. Each node discovers 2-hop neighboring information and performs a distributed election of a set of MPRs. Nodes select MPRs such that there exists a path to each of its 2-hop neighbors via a node selected as an MPR. Then, MPR nodes source and forward *Topology Control* (TC) messages that contain the MPR selectors. Note that each node independently selects its own set of MPRs.

In order to build the topology database, each node, which has been selected as MPR, periodically generates and broadcasts TC messages at a regular time interval around 5 seconds by default. Thus, TC messages are broadcasted and retransmitted by MPRs only. Upon receiving a TC message from a neighboring MPR, the receiving MPR must retransmit the message in at most 0.5 seconds.

Given the link state information acquired through periodic TC messages, the routing table for each node can be computed using the shortest path algorithm. In route calculation, the MPRs are used to form the route from a given node to any destination in the network.

OLSR has been largely adopted without modification in IWMNs, and besides, new routing protocols specifically designed for IWMNs, such as *Radio Aware Optimized Link State Routing* (RA-OLSR) [12], *Hybrid Wireless Mesh Protocol* (HWMP) [12] and *Wireless-mesh-network Proactive Routing* (WPR) [13], have adopted OLSR as a basis, including its strategy for disseminating topological information.

Consequently, as a common strategy, OLSR, RA-OLSR, HWMP and WPR propagate link-state information in a periodic basis, since their messages are transmitted without any guarantee of delivery. Thus, such protocols do not bother with reliability. They simply flood link-state information often enough to make sure that the topological databases do not remain unsynchronized for extended periods of time. Indeed, they suppose that, in some moment, all nodes receive updated topological information, and then, their topological databases can become consistent and synchronized.

In contrast, the MLSD protocol proposed herein adopts a reliable, event-based approach. Taking into account that MLSD messages are transmitted with guarantee of delivery, it does not require periodic and repetitive propagation of link-state advertisements, and so, has the potential for reducing the control overhead.

III. THE MLSD PROTOCOL

The protocol proposed herein has been developed to be adopted in the topology layer of a three-layered routing architecture, called *Infrastructure Wireless Mesh Routing Architecture* (IWMRA) [14], specifically designed taking into account IWMNs architectural features. The IWMRA architecture splits routing functionalities into 3 layers: *neighborhood*, *topology* and *routing*. The *neighborhood layer* detects the presence or absence of directly reachable neighbors. Based on a flooding approach, the *topology layer* disseminates neighborhood information all over the network. Then, the *routing layer* builds the best routes for all nodes. Consequently, a specific protocol ought to be developed for dealing with issues and functionalities in each layer.

This section presents the *Mesh Network Link State Discovery* (MLSD) protocol, a link state topology discovery protocol for IWMNs, which takes in account requirements related to scalability, robustness and reliable delivery. In order to become scalable, MLSD tries to reduce the signaling overhead for disseminating link state updates.

In order to reduce the signaling overhead for updating topology information, instead of using a periodic-based signaling strategy, MLSD adopts a *proactive, reliable, incremental, controlled and event-based signaling strategy*. In such an *event-based strategy*, a given node only sources and emits a signaling message in the event that occurs a modification in the network topology, for instance, a given mesh client (MC) establishes a new wireless link by moving within the coverage range of a given mesh router (MR).

However, taking into account the event-based strategy for fast mobile MCs, the number of messages could drastically increase as the number of events related to establishment and disconnection of wireless links among MRs and MCs intensively occurs. To deal with such an issue, MLSD adopts a *controlled strategy* for limiting the time interval between messages, and so, it tries to reduce the transmission of excessive messages in a short time interval. Consequently, considering fast mobile MCs, the controlled strategy dynamically adapts the event-based strategy to a periodic-based strategy.

To further reduce the signaling overhead, MLSD adopts an *incremental strategy* for disseminating link state information. In such a strategy, instead of propagating the whole set of link state information, MLSD propagates only updates related to modifications detected in the state of wireless links among MRs and MCs. Thus, it reduces the size of the signaling messages and consequently the signaling overhead.

Unlike other link-state routing protocols, which do not bother with reliability, MLSD adopts a *reliable strategy* for disseminating link state updates as another mechanism for reducing signaling overhead. In such a strategy, MLSD adapts the classical flooding process as a mean to implement an implicit scheme known as *positive acknowledgement with retransmission*, which guarantees reliability of flooded

 ISBN: 978-1-61208-318-6

signaling messages, ensuring consistent and synchronized topological databases without the need of repeatedly propagate the same link state information, as implemented by other link-state routing protocols.

In complement, the proposed protocol adopts a compact format for messages, grouping link state information whenever possible and eliminating outdated link state information. By acting together, all of such strategies have the potential of reducing the signaling overhead.

The MLSD protocol has been designed taking into account IWMNs architectural features. In summary, MLSD assumes that IWMNs meet the following requirements: *(i)* the set of stationary MRs provides a multi-hop wireless backbone that completely covers the interested area; *(ii)* the set of MRs are power enabled devices directly connected to an unlimited power supply; *(iii)* MCs can move or stay stationary within the wireless backbone area; and *(iv)* wireless links are bidirectional. In this initial version of the proposed protocol, each MR and MC has only one wireless network interface card.

In the following, the main concepts, strategies, and features of the MLSD protocol are presented and discussed. Initially, an overview of the proposed protocol and its operation are described. Then, the message and data structures adopted by MSLD are introduced, depicting how the protocol propagates and stores topological information. Thereafter, the strategies for disseminating link state information are detailed, describing their control mechanisms. Note that, due to space limitation, several details have to be omitted, but can be found in [15].

A. Fundamentals

The MLSD protocol implements the topology layer of the IWMRA architecture. It is responsible for disseminating link state information throughout the wireless backbone using topology update messages, which are emitted whenever occurs an event related to establishment and disconnection of wireless links among MRs and MCs. Such events are triggered by the neighborhood layer of the IWMRA architecture, which is implemented by another protocol called *Scalable Neighborhood Discovery Protocol* (SNDP) [16]. As examples of events, it can be cited a given MR adding or removing a given MC as a neighbor.

In MLSD, only MRs can source, process and broadcast topology update messages, called *Link State Updates* (LSUs). Thus, MCs cannot source, process or broadcast LSUs. MLSD manages the emission of LSUs for reducing the signaling overhead, making possible to provide better scalability by allowing a large collection of MCs in the wireless network. Besides, the proposed protocol eliminates outdated link state information, avoiding inconsistences in the topological database, which must be identical in all MRs, allowing each one to construct a complete and consistent picture of the wireless network topology.

The main MLSD contribution is the generation of messages using a reliable, incremental and event-based strategy for disseminating link state information. The consistence of the topology databases are ensured by two cooperating processes: *(i)* a reliable, incremental flooding process for disseminating link state updates, and *(ii)* a synchronization process for synchronizing topological databases in all MRs.

In order to reduce the total number of LSUs, MLSD controls the emission of messages aggregating several events in a single LSU message. In addition, it controls the time interval between consecutive LSUs by managing and delaying the dissemination of events related to fast mobile MCs. In complement, by adopting a compact format for representing link state operations, it also reduces the signaling overhead transported in LSUs.

B. Message Structure

Each *Link State Update* (LSU) is a packet employed by MLSD for disseminating link state updates in the wireless backbone. As illustrated in Fig. 1, each LSU can carry one or several announcements related to events that occur in the network topology. In MSLD, such announcements are called *Link State Advertisements* (LSAs). In turn, each LSA can carry one or several *Link State Operations* (LSOs) that occur in a given MR, represented by the establishment or disconnection of wireless links with other MRs or MCs.

When generated, an LSU message is directly encapsulated in a frame of the data link or *Media Access Control* (MAC) layer and then transmitted in broadcast. All MRs that receive an LSU must evaluate each encapsulated LSA, and then decide whether or not to process or forward (or both) each encapsulated LSO.

In complement, each LSO can require an additional processing taking into account the type of neighboring node (MR or MC) involved in the respective link state update. On the one hand, if the involved node is a MC, such a processing is always the simple addition or removal of the wireless link in the topological database. On the other hand, if the involved node is another MR, such a processing can also initiate a synchronization process for synchronizing topological databases or a garbage collection process for removing all unreachable nodes from the topological database.

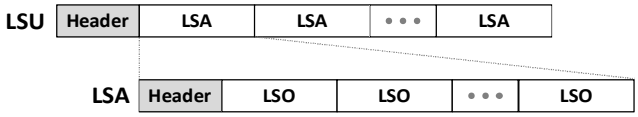


Figure 1. Message structure.

C. Send Buffer

The send buffer is an internal data structure adopted by all MRs for storing LSOs and their related information. Considering a given MR, all LSOs present in its send buffer can have the following origin: *(i)* auto-generated by the own MR, considering link state updates with its neighboring nodes; and *(ii)* received in LSUs generated or forwarded by other neighboring MRs. Note that, all LSOs in the send buffer must be disseminated in broadcast for all MRs in the wireless backbone. Therefore, LSOs in the send buffer constitute the base for creating LSAs and then LSUs, which are flooded in the wireless backbone.

 ISBN: 978-1-61208-318-6

Each LSO in the send buffer is classified as: *(i)* a new operation that still needs to be forwarded to all neighboring MRs; or *(ii)* an already-transmitted operation that needs to be confirmed by or retransmitted to neighboring MRs. Besides, MLSD also keeps timers that indicate when the LSO must be transmitted for the first time or retransmitted, if at all need.

The retransmission of a given LSO is required whenever the delivery control mechanism detects that one or several neighboring MRs did not acknowledge the receipt of the previous transmission of the LSO. Such a delivery control mechanism is implemented by defining a list of forwarders associated with each LSO in the send buffer. The list of forwarders represents all neighboring MRs that still need to receive and forward the respective LSO.

D. Topological Database

The topological database represents a map of the wireless network topology. It is important to emphasize that the topological database is employed by the routing layer of the IWMRA architecture for proactively calculating all routes between each pair of nodes. In a given MR, its topological database stores link state information, which are directly auto-generated by the neighborhood layer of the own MR or received in LSUs propagated by other neighboring MRs.

It is important to note that link state information in the topological database does not expire ever. Such information can only be included or removed through LSOs propagated by the MLSD protocol. The topological database must be identical, consistent and synchronized in all MRs that compose the wireless backbone. Thus, all MRs know the state of the wireless links defined among all MRs and MCs in the wireless network.

MLSD also adopts a versioning scheme for LSOs. In such a scheme, each LSO generated by a given MR has a unique sequence number, which is managed and assigned by the initial source MR. Upon receiving a given LSO, the receiving MR substitutes the old version of the LSO in its topological database with the new one.

E. Link-State Propagation

As already indicated, the MLSD protocol keeps consistent topological databases exploring two independent but cooperating processes: *(i)* a reliable, incremental flooding process for disseminating link state updates, and *(ii)* a synchronization process for synchronizing topological databases in all MRs.

The dissemination of link state updates integrated in the MLSD protocol adopts the well-known concept of flooding. The flooding process is performed via LSUs, which contain incremental updates of the network topology in encapsulated LSAs, which in turn encapsulate LSOs. Such LSOs, when processed by a given MR, are forwarded to its neighboring MRs, until reaching hop-by-hop all MRs in the wireless backbone.

In order to perform the reliable delivery, the flooding process implements an implicit scheme known as positive acknowledgement with retransmission. Besides, when processing LSU messages, it determines the type of action to be taken for each encapsulated LSO, which can be to forward, acknowledge, retransmit or simply ignore the LSO. Thus, the flooding process allows disseminating link state updates throughout the wireless backbone and, together with the versioning scheme, also ensures that old versions of LSOs, identified by their sequence numbers, do not affect the consistence of the topological databases. In complement, MLSD manages the time interval between retransmissions, avoiding the excess of messages triggered by topological events in the wireless network.

Taking into account the implicit, positive acknowledgement with retransmission scheme, a given *MR-X* that propagates an LSU detects the effective reception of each encapsulated LSO by all neighboring MRs, indicated as the list of forwarders for each LSO in the send buffer, when they forward the same LSOs in their own LSUs. Since all LSUs are transmitted in broadcast, *MR-X* also receives the LSUs from its neighbors, and so, such LSUs can serve as delivery acknowledgements from the forwarders to *MR-X*.

If a given forwarding *MR-F* does not transmit a given LSO within a specified time interval, *MR-X* retransmits the LSO again, until detecting that *MR-F* has forwarded it. Upon detecting that all forwarders have received the LSO, internally, *MR-X* declares the successful forwarding of the LSO, removing it from the send buffer.

Fig. 2 illustrates the implicit, positive acknowledgement scheme. In Fig. 2a, *MR-A* sends an LSU that contains an LSO to its unique forwarder *MR-B*, indicating the establishment of a wireless link between *MR-A* and *MC-X*. In turn, *MR-B* forwards the LSO in another LSU, indicating *MR-C* as a forwarder (Fig. 2b). In this case, note that, the LSU from *MR-B* is received by both *MR-A* and *MR-C*. On the one hand, *MR-A* interprets the LSU from *MR-B* as an acknowledgement. On the other hand, since *MR-C* has been indicated as forwarder by *MR-B*, it must forward the LSO. Though, *MR-C* does not have neighbors to forward the message. Even thus, *MR-C* transmits the LSO with no forwarders, allowing *MR-B* to acknowledge that *MR-C* has successfully received the LSO (Fig. 2c).

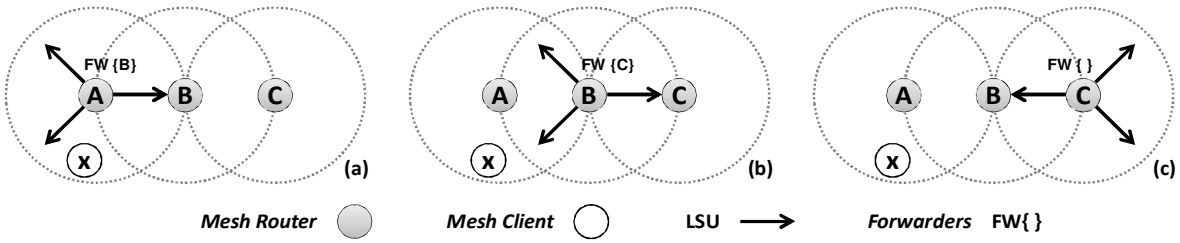


Figure 2. Implicit, positive acknowledgement mechanism.

 ISBN: 978-1-61208-318-6

It is important to stress that no extra message needs to be sent since the acknowledgement is implicit. Thus, when LSOs are successfully received and forwarded, each MR needs transmit each LSO only one time. Such a strategy is different from the flooding process employed by the OLSR protocol, in which topology control messages are also transmitted only one time by each node, however, since OLSR does not provide any guarantee of delivery, messages have to be periodically retransmitted.

In order to avoid collisions between neighboring MRs that forward link state updates in the flooding process, and so retransmissions of LSUs, MLSD adopts a *time-slot based strategy*. In such a strategy, time slots are time intervals in which MRs are allowed to transmit LSUs. Considering a given transmitted LSU, all forwarding MRs configure their time slots according to the position of each MR in the list of forwarders indicated in the LSU. Thus, each LSU indicates the specific and distinct time slot for each forwarding MR.

Note that the MAC layer of IEEE 802.11 wireless networks adopts the *Distributed Coordination Function* (DCF) with a contention window for dealing with collisions. However, in MLSD, upon receiving an LSU, all forwarding MRs would try at the same time to forward the encapsulated LSOs, and so, as widely known, DCF can lead to collisions in situations in which many nodes attempt to communicate at the same time. Thus, in a more efficient way, the time-slot mechanism distributes in different time slots the instant at which each forwarding MR tries to transmit.

In a given MR, upon receiving an LSU, the forwarding of the encapsulated LSOs is delayed by taking into account the time slot allocated to the forwarding MR in the LSU. As mentioned, the time slots are calculated using the position of the MR in the list of forwarders indicated in the LSU. Thus, the first MR in the list forwards in the first slot, the second MR in the second slot, and so on. Note that the time-slot mechanism acts together with the 802.11 DCF, but now, forwarding MRs do not try at the same time to send LSOs, avoiding collisions that could not be avoided by DCF only.

F. Synchronizing Topological Databases

Whenever an MR is initialized in the wireless backbone, it needs to create a topological database, which ought to have all link state information already stored in other MRs. Consequently, once a given MR detects as neighbor another MR, they must perform the synchronization of their topological databases.

Such a synchronization process is initiated by the MR that firstly detects the establishment of the wireless link with another neighboring MR. In the initial phase, the detecting MR assemblies all LSOs based on its topological database, and then inserts them in the send buffer, indicating as forwarder the recently discovered neighboring MR. The assembly of LSOs is possible because the topological database stores the original sequence number of each stored link state information.

LSOs associated with the synchronization process only have operations for adding wireless links between MRs and MCs, since no information is stored in the topological database about already disconnected wireless links.

In the synchronization process, the dissemination of LSOs follows the same procedures and rules adopted in the flooding process. It is important to emphasize that, upon receiving LSOs, the forwarding MR evaluates each one and decides what to do according to the implicit, positive acknowledgement with retransmission scheme. For instance, the forwarding MR can decide to retransmit as an acknowledgement. However, in case of partitioned backbone, it has to forward to other neighboring MRs.

Fig. 3 depicts the synchronization process. When *MR-B* detects the wireless link with *MR-A*, it inserts in its send buffer all LSOs related to link state information stored in its topological database, indicating *MR-A* as a forwarder. Then, *MR-B* sends one or more LSUs encapsulating such LSOs (Fig. 3a). Thus, upon receiving LSUs, *MR-A* confirms *MR-B* as neighbor, if at all needed. Then, *MR-A* inserts in its send buffer LSOs related to link state information stored in its own topological database and also LSOs received from *MR-B* that need to be acknowledged. Thereafter, *MR-A* sends one or more LSUs encapsulating such LSOs (Fig. 3b). After receiving LSUs, *MR-B* declares as successful the transmission of its LSOs to *MR-A*, and besides, it sends acknowledgements for all LSOs received from *MR-A* (Fig. 4c). In conclusion, *MR-A* declares as successful the transmission of its LSOs to *MR-B*.

IV. Performance Evaluation

The signaling overhead generated by topology discovery protocols has a strong impact on the performance of the routing protocols [17]. Thus, in order to evince the MLSD performance gains, a simulation-based performance evaluation has been conducted using the NS-2 simulator [18], contrasting MLSD and the OLSR topology discovery process. Note that other protocols for IWMNs, including RA-OLSR [12], HWMP [12] and WPR [13], also adopt topology discovery processes similar to OLSR. Thus, it makes sense to contrast MLSD against the OLSR topology discovery process, which is the basis for all other ones.

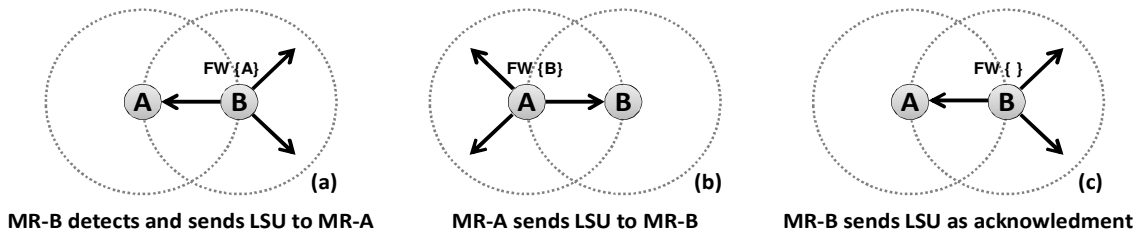


Figure 3. Synchronizing topological databases.

 ISBN: 978-1-61208-318-6

The efficiency of MLSD has been evaluated in several simulated scenarios, varying the number and the speed of devices. As a way to show the general MLSD behavior, this paper presents the performance gains in scenarios defined by a grid of 10x10 stationary MRs, in which up to 100 mobile MCs adopt an average speed of 10 m/s, varying uniformly between 0 and 20 m/s. For each scenario, average values of the signaling overhead are calculated based on several simulation experiments, considering a relative estimation error of 5% and a confidence interval of 95%. Each experiment has a simulation time of 3.000 seconds, from which the first 160 seconds are discarded as an initial transient. Interested readers can find in [15] a detailed description of simulation settings, scenarios and outcomes.

In the context of IWMNs, as illustrated in Fig. 4, simulation results make clear that MLSD is a better option than the OLSR topology discovery process in terms of signaling overhead in bytes. In Fig. 4, it is possible to note that OLSR suffers more influence from the increase in the quantity of mobile MCs in the wireless network. The poor behavior of the OLSR protocol is mainly influenced by its periodic-based strategy, adopted by MRs and also MCs for disseminating link state information through their MPRs. In contrast, the excellent MLSD behavior is a direct consequence of the combination of its controlled, event-based strategy, in which only MRs disseminate LSUs in the event that occur modifications in the network topology.

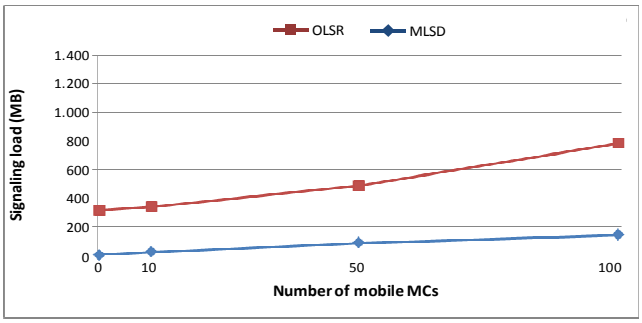


Figure 4. Signaling overhead

In Fig. 4, considering 0, 10 and 50 MCs, the event-based strategy makes possible MLSD to have a significant smaller signaling overhead, but that grows in a comparable way with OLSR. But, from 50 to 100 MCs, the controlled strategy begins to act in MLSD when the frequency of link state updates increases as a whole, and consequently the signaling overhead for MLSD has a growth smaller than OLSR.

V. CONCLUSION

This paper has proposed MLSD, a network topology discovery protocol based on the link-state approach and specifically designed for IWMNs. Regarding the signaling overhead, MLSD has an excellent behavior in typical IWMNs, becoming much more scalable than the OLSR topology discovery process. Thus, it is possible to guess that MLSD has the potential to become a better choice than the IEEE 802.11s proposal for mesh networks, in which RA-OLSR adopts the OLSR topology discovery process.

Despite the interesting outcomes in terms of signaling overhead, as a future work, it is still needed to evaluate MLSD in relation to other performance metrics. For instance, it is under laboratory work the evaluation of the convergence time for the topological database, which can reveal the time interval required for synchronizing link-sate information in all nodes. In pilot investigations, considering the reliable strategy adopted by MLSD for disseminating link-state updates, it is expected to confirm that MLSD also has a behavior better than OLSR in terms of convergence time. Also, as known, OLSR has problems with topological database convergence, which do not occur in MLSD.

REFERENCES

[1] I. F. Akyildiz, and W. Xudong, “A survey on wireless mesh networks”, Communications Magazine, IEEE, v.43, n.9, 2005, pp. S23-S30.

[2] Y. Zhang, J. Luo, and H. Hu, “Wireless mesh networking: architectures, protocols and standards”, Auerbach Pub, 2006.

[3] E. Hossain, and K. K. Leung, “Wireless mesh networks: architectures and protocols”, Springer, 2008.

[4] N. Tsarmpopoulos, I. Kalavros, and S. Lalis, “A low-cost and simple-to-deploy peer-to-peer wireless network based on open source Linux routers”. 1st Int. Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005, pp. 92-97.

[5] T. Clausen, and P. Jacquet, “RFC3626 - Optimized Link State Routing Protocol (OLSR)”, IETF, 2003.

[6] Microsoft Mesh Networks. Available from: <http://research.microsoft.com/mesh/> [retrieved: 12, 2013].

[7] D. Johnson, D. Maltz, and Y. Hu, “RFC4728 - The Dynamic Source Routing protocol (DSR) for mobile ad hoc networks for IPv4”, IETF, 2007.

[8] A. Nezhad, A. Miri, and D. Makrakis, “Efficient topology discovery for multihop wireless sensor networks”, IEEE Computer Society, 2008, pp. 358-365.

[9] X. Zou, B. Ramamurthy, and S. Magliveras, “Routing techniques in wireless ad hoc networks classification and comparison”, 6th World Multiconference on Systemics, Cybernetics and Informatics, 2002, pp. 1-6.

[10] J. Chen, *et al.*, “Performance comparison of AODV and OFLSR in wireless mesh networks”, IFIP Mediterranean Ad Hoc Networking Workshop, 2006, pp. 271-278.

[11] C. E. Perkins, and E. M. Royer, “Ad-hoc on-demand distance vector routing”, 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90-100.

[12] M. Bahr, “Proposed routing for IEEE 802.11s WLAN mesh networks”, 2nd Int. Workshop on Wireless Internet, 2006.

[13] M. E. Campista, L. Costa, and O. Duarte, “WPR: a proactive routing protocol tailored to wireless mesh networks”, IEEE Global Communicattions Conference, 2008, pp. 538-542.

[14] D. C. Porto, G. Cavalcanti, and G. Elias, “A layered routing architecture for infrastructure wireless mesh networks”, 5th Int. Conf. on Networking and Services, 2009, pp. 366-369.

[15] D. C. Porto, “MLSD: a link-state topology discovery protocol for infrastructure wireless mesh networks”, Master Thesis, Federal University of Paraíba, 2010 (in portuguese).

[16] G. Elias, M. Novaes, G. Cavalcanti, D. C. Porto, “Simulation-based performance evaluation of the SNDP protocol for infrastructure WMNs”, 24th IEEE Int. Conf. on Advanced Information Networking and Applications, 2010, pp. 90-97.

[17] F. J. Ros, “UM-OLSR”. Available from: <http://masimum.inf.um.es/fjrm/development/um-olsr/> [retrieved: 12, 2013].

[18] K. Fall, and K. Varadhan, “The ns manual”, The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC, 2011.

 ISBN: 978-1-61208-318-6

Coverage and Lifetime Optimization in Heterogeneous Energy Wireless Sensor Networks

Ali Kadhum Idrees, Karine Deschinkel, Michel Salomon, and Raphaël Couturier
FEMTO-ST Institute, UMR 6174 CNRS
University of Franche-Comté
Belfort, France
Email: ali.idness@edu.univ-fcomte.fr, {karine.deschinkel, michel.salomon, raphael.couturier}@univ-fcomte.fr

***Abstract*—One of the fundamental challenges in Wireless Sensor Networks (WSNs) is the coverage preservation and the extension of the network lifetime continuously and effectively when monitoring a certain area (or region) of interest. In this paper, a coverage optimization protocol to improve the lifetime in heterogeneous energy wireless sensor networks is proposed. The area of interest is first divided into subregions using a divide-and-conquer method and then the scheduling of sensor node activity is planned for each subregion. The proposed scheduling considers rounds during which a small number of nodes, remaining active for sensing, is selected to ensure coverage. Each round consists in four phases: (i) Information Exchange, (ii) Leader Election, (iii) Decision, and (iv) Sensing. The decision process is carried out by a leader node, which solves an integer program. Simulation results show that the proposed approach can prolong the network lifetime and improve the coverage performance.**

***Keywords*-Wireless Sensor Networks; Area Coverage; Network Lifetime; Optimization; Scheduling.**

I. INTRODUCTION

Recent years have witnessed significant advances in wireless communications and embedded micro-sensing Micro-Electro-Mechanical Systems (MEMS) technologies which have led to the emergence of Wireless Sensor Networks (WSNs) as one of the most promising technologies [1]. In fact, they present huge potential in several domains ranging from health care applications to military applications. A sensor network is composed of a large number of tiny sensing devices deployed in a region of interest. Each device has processing and wireless communication capabilities, which enable it to sense its environment, to compute, to store information and to deliver report messages to a base station [2]. One of the main design issues in WSNs is to prolong the network lifetime, while achieving acceptable quality of service for applications. Indeed, sensors nodes have limited resources in terms of memory, energy and computational power.

Since sensor nodes have limited battery life; since it is impossible to replace batteries, especially in remote and hostile environments, it is desirable that a WSN should be deployed with high density because spatial redundancy can then be exploited to increase the lifetime of the network. In such a high density network, if all sensor nodes were to be activated at the same time, the lifetime would be reduced. To extend the lifetime of the network, the main idea is to take advantage of the overlapping sensing regions of some sensor nodes to save energy by turning off some of them during the sensing phase [3]. Obviously, the deactivation of nodes is only relevant if the coverage of the monitored area is not affected. In this paper, we concentrate on the area coverage problem [4], with the objective of maximizing the network lifetime by using an adaptive scheduling. The area of interest is divided into subregions and an activity scheduling for sensor nodes is planned for each subregion. In fact, the nodes in a subregion can be seen as a cluster where each node sends sensing data to the cluster head or the sink node. Furthermore, the activities in a subregion/cluster can continue even if another cluster stops due to too many node failures. Our scheduling scheme considers rounds, where a round starts with a discovery phase to exchange information between sensors of the subregion, in order to choose in a suitable manner a sensor node to carry out a coverage strategy. This coverage strategy involves the solving of an integer program, which provides the activation of the sensors for the sensing phase of the current round.

The remainder of the paper is organized as follows. The next section reviews the related work in the field. Section III is devoted to the scheduling strategy for energy-efficient coverage. Section IV gives the coverage model formulation, which is used to schedule the activation of sensors. Section V shows the simulation results obtained using the discrete event simulator OMNeT++ [5]. They fully demonstrate the usefulness of the proposed approach. Finally, we give concluding remarks and some suggestions for future works in Section VI.

II. RELATED WORKS

In this section, we only review some recent works dealing with the coverage lifetime maximization problem, where the objective is to optimally schedule sensors' activities in order to extend WSNs lifetime. Vu [6] proposed a novel distributed heuristic, called Distributed Energy-efficient Scheduling for k-coverage (DESK), which ensures that the energy consumption among the sensors is balanced and the lifetime maximized while the coverage requirement is maintained. This heuristic works in rounds, requires only 1-hop neighbor information, and each sensor decides its status (active or sleep) based on the perimeter coverage model proposed by Huang and Tseng [7]. More recently, Shibo et al. [8] expressed the coverage problem as a minimum weight submodular set cover problem and proposed a Distributed Truncated Greedy Algorithm (DTGA) to solve it. They take, in particular, advantage from

 ISBN: 978-1-61208-318-6

both temporal and spatial correlations between data sensed by different sensors.

The works presented in [9], [10], [11] focus on the definition of coverage-aware, distributed energy-efficient and distributed clustering methods respectively. They aim to extend the network lifetime while ensuring the coverage. S. Misra et al. [3] proposed a localized algorithm which conserves energy and coverage by activating the subset of sensors with the minimum overlapping area. It preserves the network connectivity thanks to the formation of the network backbone. J. A. Torkestani [12] designed a Learning Automata-based Energy-Efficient Coverage protocol (LAEEC) to construct a Degree-constrained Connected Dominating Set (DCDS) in WSNs. He showed that the correct choice of the degree-constraint of DCDS balances the network load on the active nodes and leads to enhance the coverage and network lifetime.

The main contribution of our approach addresses three main questions to build a scheduling strategy.

How must the phases for information exchange, decision and sensing be planned over time? Our algorithm divides the timeline into rounds. Each round contains 4 phases: Information Exchange, Leader Election, Decision, and Sensing.

What are the rules to decide which node has to be turned on or off? Our algorithm tends to limit the overcoverage of points of interest to avoid turning on too many sensors covering the same areas at the same time, and tries to prevent undercoverage. The decision is a good compromise between these two conflicting objectives.

Which node should make such a decision? A leader node should make such a decision. Our work does not consider only one leader to compute and to broadcast the scheduling decision to all the sensors. When the network size increases, the network is divided into many subregions and the decision is made by a leader in each subregion.

III. ACTIVITY SCHEDULING

We consider a randomly and uniformly deployed network consisting of static wireless sensors. The wireless sensors are deployed in high density to ensure initially a full coverage of the interested area. We assume that all nodes are homogeneous in terms of communication and processing capabilities and heterogeneous in term of energy provision. The location information is available to the sensor node either through hardware such as embedded GPS or through location discovery algorithms. The area of interest can be divided using the divide-and-conquer strategy into smaller areas called subregions and then our coverage protocol will be implemented in each subregion simultaneously. Our protocol works in rounds fashion, as shown in Figure 1.

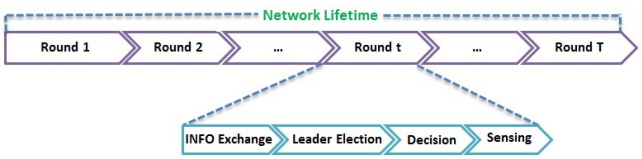


Figure 1. Multi-Round Coverage Protocol.

Each round is divided into 4 phases: Information (INFO) Exchange, Leader Election, Decision, and Sensing. For each round, there is exactly one set cover responsible for the sensing task. This protocol is more reliable against an unexpected node failure because it works in rounds. On the one hand, if a node failure is detected before making the decision, the node will not participate to this phase, and, on the other hand, if the node failure occurs after the decision, the sensing task of the network will be temporarily affected: only during the period of sensing until a new round starts, since a new set cover will take charge of the sensing task in the next round. The energy consumption and some other constraints can easily be taken into account since the sensors can update and then exchange their information (including their residual energy) at the beginning of each round. However, the pre-sensing phases (INFO Exchange, Leader Election, Decision) are energy consuming for some nodes, even when they do not join the network to monitor the area. Below, we describe each phase in more details.

A. Information exchange phase

Each sensor node j sends its position, remaining energy RE_j, and the number of local neighbours NBR_j to all wireless sensor nodes in its subregion by using an INFO packet and then listens to the packets sent from other nodes. After that, each node will have information about all the sensor nodes in the subregion. In our model, the remaining energy corresponds to the time that a sensor can live in the active mode.

B. Leader election phase

This step includes choosing the Wireless Sensor Node Leader (WSNL), which will be responsible for executing the coverage algorithm. Each subregion in the area of interest will select its own WSNL independently for each round. All the sensor nodes cooperate to select WSNL. The nodes in the same subregion will select the leader based on the received information from all other nodes in the same subregion. The selection criteria in order of priority are: larger number of neighbours, larger remaining energy, and then in case of equality, larger index.

C. Decision phase

The WSNL will solve an integer program (see section IV) to select which sensors will be activated in the following sensing phase to cover the subregion. WSNL will send Active-Sleep packet to each sensor in the subregion based on the algorithm's results.

D. Sensing phase

Active sensors in the round will execute their sensing task to preserve maximal coverage in the region of interest. We will assume that the cost of keeping a node awake (or asleep) for sensing task is the same for all wireless sensor nodes in the network. Each sensor will receive an Active-Sleep packet from WSNL informing it to stay awake or to go to sleep for a time equal to the period of sensing until starting a new round.

We consider a boolean disk coverage model which is the most widely used sensor coverage model in the literature. Each sensor has a constant sensing range R_s. All space points within a disk centered at the sensor with the radius of the sensing

 ISBN: 978-1-61208-318-6

range is said to be covered by this sensor. We also assume that the communication range $R_c \geq 2R_s$ [13].

Instead of working with the coverage area, we consider for each sensor a set of points called primary points. We also assume that the sensing disk defined by a sensor is covered if all the primary points of this sensor are covered. By knowing the position (point center: (p_x, p_y)) of a wireless sensor node and its R_s, we calculate the primary points directly based on the proposed model. We use these primary points (that can be increased or decreased if necessary) as references to ensure that the monitored region of interest is covered by the selected set of sensors, instead of using all the points in the area.

We can calculate the positions of the selected primary points in the circle disk of the sensing range of a wireless sensor node (see figure 2) as follows:
(p_x, p_y) = point center of wireless sensor node
$X_1 = (p_x, p_y)$
$X_2 = (p_x + R_s * (1), p_y + R_s * (0))$
$X_3 = (p_x + R_s * (-1), p_y + R_s * (0))$
$X_4 = (p_x + R_s * (0), p_y + R_s * (1))$
$X_5 = (p_x + R_s * (0), p_y + R_s * (-1))$
$X_6 = (p_x + R_s * (\frac{-\sqrt{2}}{2}), p_y + R_s * (0))$
$X_7 = (p_x + R_s * (\frac{\sqrt{2}}{2}), p_y + R_s * (0))$
$X_8 = (p_x + R_s * (\frac{-\sqrt{2}}{2}), p_y + R_s * (\frac{-\sqrt{2}}{2}))$
$X_9 = (p_x + R_s * (\frac{\sqrt{2}}{2}), p_y + R_s * (\frac{-\sqrt{2}}{2}))$
$X_{10} = (p_x + R_s * (\frac{-\sqrt{2}}{2}), p_y + R_s * (\frac{\sqrt{2}}{2}))$
$X_{11} = (p_x + R_s * (\frac{\sqrt{2}}{2}), p_y + R_s * (\frac{\sqrt{2}}{2}))$
$X_{12} = (p_x + R_s * (0), p_y + R_s * (\frac{\sqrt{2}}{2}))$
$X_{13} = (p_x + R_s * (0), p_y + R_s * (\frac{-\sqrt{2}}{2}))$.

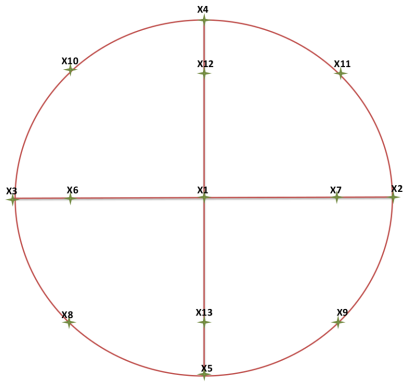


Figure 2. Sensor node represented by 13 primary points.

IV. Coverage problem formulation

Our model is based on the model proposed by Pedraza et al. [14], where the objective is to find a maximum number of disjoint cover sets. To accomplish this goal, authors proposed an integer program, which forces undercoverage and overcoverage of targets to become minimal at the same time. They use binary variables x_{jl} to indicate if sensor j belongs to cover set l. In our model, we consider binary variables X_j, which determine the activation of sensor j in the sensing phase of the round. We also consider primary points as targets. The set of primary points is denoted by P and the set of sensors by J.

For a primary point p, let α_{jp} denote the indicator function of whether the point p is covered, that is:

$$\alpha_{jp} = \begin{cases} 1 & \text{if the primary point } p \text{ is covered} \\ & \text{by sensor node } j, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The number of active sensors that cover the primary point p is equal to $\sum_{j \in J} \alpha_{jp} * X_j$ where:

$$X_j = \begin{cases} 1 & \text{if sensor } j \text{ is active,} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We define the Overcoverage variable Θ_p as:

$$\Theta_p = \begin{cases} 0 & \text{if the primary point} \\ & p \text{ is not covered,} \\ \left(\sum_{j \in J} \alpha_{jp} * X_j\right) - 1 & \text{otherwise.} \end{cases} \quad (3)$$

More precisely, Θ_p represents the number of active sensor nodes minus one that cover the primary point p.
The Undercoverage variable U_p of the primary point p is defined by:

$$U_p = \begin{cases} 1 & \text{if the primary point } p \text{ is not covered,} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Our coverage optimization problem can then be formulated as follows

$$\begin{cases} \min \sum_{p \in P} (w_\theta \Theta_p + w_U U_p) & \\ \text{subject to :} & \\ \sum_{j \in J} \alpha_{jp} X_j - \Theta_p + U_p = 1, & \forall p \in P \\ \Theta_p \in \mathbb{N}, & \forall p \in P \\ U_p \in \{0,1\}, & \forall p \in P \\ X_j \in \{0,1\}, & \forall j \in J \end{cases} \quad (5)$$

- X_j : indicates whether or not the sensor j is actively sensing in the round (1 if yes and 0 if not);
- Θ_p : *overcoverage*, the number of sensors minus one that are covering the primary point p;
- U_p : *undercoverage*, indicates whether or not the primary point p is being covered (1 if not covered and 0 if covered).

The first group of constraints indicates that some primary point p should be covered by at least one sensor and, if it is not always the case, overcoverage and undercoverage variables help balancing the restriction equations by taking positive values. There are two main objectives. Firstly, we limit the overcoverage of primary points in order to activate a minimum number of sensors. Secondly, we prevent the absence of monitoring on some parts of the subregion by minimizing the undercoverage. The weights w_θ and w_U must be properly chosen so as to guarantee that the maximum number of points are covered during each round.

 ISBN: 978-1-61208-318-6

V. SIMULATION RESULTS

In this section, we conducted a series of simulations to evaluate the efficiency and the relevance of our approach, using the discrete event simulator OMNeT++ [5]. We performed simulations for five different densities varying from 50 to 250 nodes. Experimental results were obtained from randomly generated networks in which nodes are deployed over a $(50 \times 25)\ m^2$ sensing field. More precisely, the deployment is controlled at a coarse scale in order to ensure that the deployed nodes can fully cover the sensing field with the given sensing range. 10 simulation runs are performed with different network topologies for each node density. The results presented hereafter are the average of these 10 runs. A simulation ends when all the nodes are dead or the sensor network becomes disconnected (some nodes may not be able to send, to a base station, an event they sense).

Our proposed coverage protocol uses the radio energy dissipation model defined by Heinzelman et al. [15] as energy consumption model for each wireless sensor node when transmitting or receiving packets. The energy of each node in a network is initialized randomly within the range 24-60 joules, and each sensor node will consume 0.2 watts during the sensing period, which will last 60 seconds. Thus, an active node will consume 12 joules during the sensing phase, while a sleeping node will use 0.002 joules. Each sensor node will not participate in the next round if its remaining energy is less than 12 joules. In all experiments, the parameters are set as follows: $R_s = 5\ m$, $w_\Theta = 1$, and $w_U = |P^2|$.

We evaluate the efficiency of our approach by using some performance metrics such as: coverage ratio, number of active nodes ratio, energy saving ratio, energy consumption, network lifetime, execution time, and number of stopped simulation runs. Our approach called strategy 2 (with two leaders) works with two subregions, each one having a size of $(25 \times 25)\ m^2$. Our strategy will be compared with two other approaches. The first one, called strategy 1 (with one leader), works as strategy 2, but considers only one region of $(50 \times 25)\ m^2$ with only one leader. The other approach, called Simple Heuristic, consists in uniformly dividing the region into squares of $(5 \times 5)\ m^2$. During the decision phase, in each square, a sensor is randomly chosen, it will remain turned on for the coming sensing phase.

A. The impact of the number of rounds on the coverage ratio

In this experiment, the coverage ratio measures how much the area of a sensor field is covered. In our case, the coverage ratio is regarded as the number of primary points covered among the set of all primary points within the field. Figure 3 shows the impact of the number of rounds on the average coverage ratio for 150 deployed nodes for the three approaches. It can be seen that the three approaches give similar coverage ratios during the first rounds. From the 9th round the coverage ratio decreases continuously with the simple heuristic, while the two other strategies provide superior coverage to 90% for five more rounds. Coverage ratio decreases when the number of rounds increases due to dead nodes. Although some nodes are dead, thanks to strategy 1 or 2, other nodes are preserved to ensure the coverage. Moreover, when we have a dense sensor network, it leads to maintain the full coverage for a larger number of rounds. Strategy 2 is slightly more efficient than strategy 1, because strategy 2 subdivides the region into 2 subregions and if one of the two subregions becomes disconnected, the coverage may be still ensured in the remaining subregion.

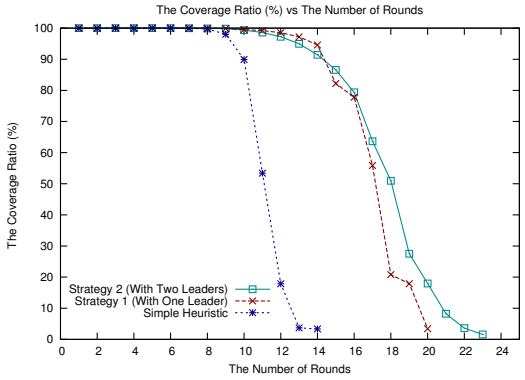


Figure 3. The impact of the number of rounds on the coverage ratio for 150 deployed nodes.

B. The impact of the number of rounds on the active sensors ratio

It is important to have as few active nodes as possible in each round, in order to minimize the communication overhead and maximize the network lifetime. This point is assessed through the Active Sensors Ratio (ASR), which is defined as follows:

$$\text{ASR}(\%) = \frac{\text{Number of active sensors during the current sensing phase}}{\text{Total number of sensors in the network for the region}} \times 100.$$

Figure 4 shows the average active nodes ratio versus rounds for 150 deployed nodes.

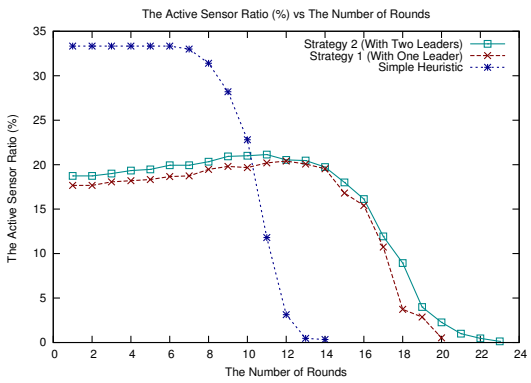


Figure 4. The impact of the number of rounds on the active sensors ratio for 150 deployed nodes.

The results presented in figure 4 show the superiority of both proposed strategies, the strategy with two leaders and the one with a single leader, in comparison with the simple heuristic. The strategy with one leader uses less active nodes than the strategy with two leaders until the last rounds, because it uses central control on the whole sensing field. The advantage of the strategy 2 approach is that even if a network is disconnected in one subregion, the other one usually continues the optimization process, and this extends the lifetime of the network.

 ISBN: 978-1-61208-318-6

C. Impact of the number of rounds on the energy saving ratio

In this experiment, we consider a performance metric linked to energy. This metric, called Energy Saving Ratio (ESR), is defined by:

$$\text{ESR}(\%) = \frac{\text{Number of alive sensors during this round}}{\text{Total number of sensors in the network for the region}} \times 100.$$

The longer the ratio is, the more redundant sensor nodes are switched off, and consequently the longer the network may live. Figure 5 shows the average Energy Saving Ratio versus rounds for all three approaches and for 150 deployed nodes.

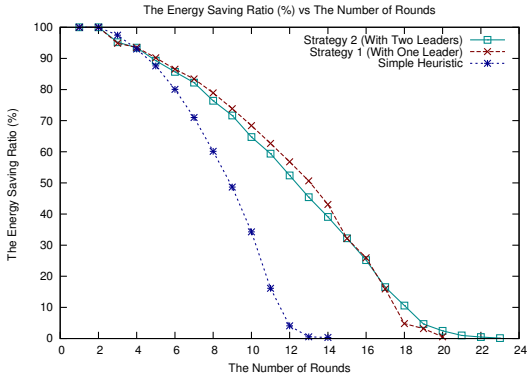


Figure 5. The impact of the number of rounds on the energy saving ratio for 150 deployed nodes.

The simulation results show that our strategies allow to efficiently save energy by turning off some sensors during the sensing phase. As expected, the strategy with one leader is usually slightly better than the second strategy, because the global optimization permits to turn off more sensors. Indeed, when there are two subregions more nodes remain awake near the border shared by them. Note that again as the number of rounds increases the two leaders' strategy becomes the most performing one, since it takes longer to have the two subregion networks simultaneously disconnected.

D. The percentage of stopped simulation runs

We will now study the percentage of simulations, which stopped due to network disconnections per round for each of the three approaches. Figure 6 illustrates the percentage of stopped simulation runs per round for 150 deployed nodes. It can be observed that the simple heuristic is the approach, which stops first because the nodes are randomly chosen. Among the two proposed strategies, the centralized one first exhibits network disconnections. Thus, as previously explained, in case of the strategy with several subregions the optimization effectively continues as long as a network in a subregion is still connected. This longer partial coverage optimization participates in extending the network lifetime.

E. The energy consumption

In this experiment, we study the effect of the multi-hop communication protocol on the performance of the strategy with two leaders and compare it with the other two approaches. The average energy consumption resulting from wireless communications is calculated by taking into account the energy spent by all the nodes when transmitting and receiving packets during the network lifetime. This average value, which is obtained for 10 simulation runs, is then divided by the average

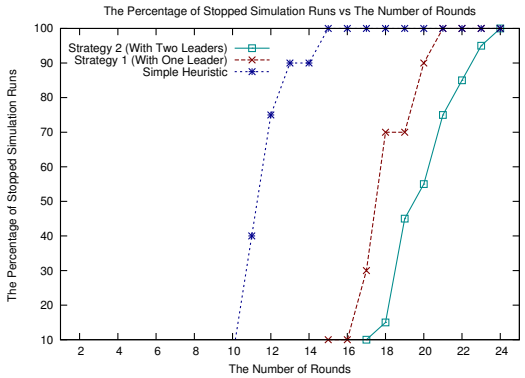


Figure 6. The percentage of stopped simulation runs compared to the number of rounds for 150 deployed nodes.

number of rounds to define a metric allowing a fair comparison between networks having different densities.

Figure 7 illustrates the energy consumption for the different network sizes and the three approaches. The results show that the strategy with two leaders is the most competitive from the energy consumption point of view. A centralized method, like the strategy with one leader, has a high energy consumption due to many communications. In fact, a distributed method greatly reduces the number of communications thanks to the partitioning of the initial network in several independent subnetworks. Let us notice that even if a centralized method consumes far more energy than the simple heuristic, since the energy cost of communications during a round is a small part of the energy spent in the sensing phase, the communications have a small impact on the network lifetime.

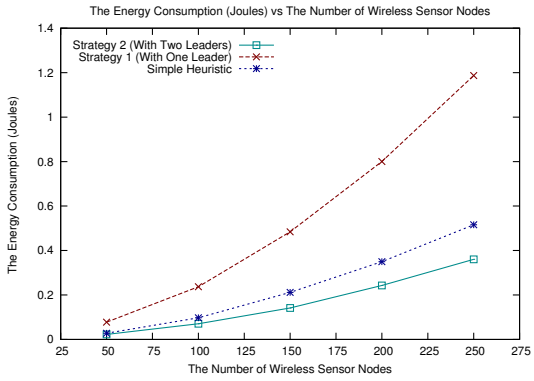


Figure 7. The energy consumption.

F. The impact of the number of sensors on execution time

A sensor node has limited energy resources and computing power, therefore it is important that the proposed algorithm has the shortest possible execution time. The energy of a sensor node must be mainly used for the sensing phase, not for the pre-sensing ones. Table I gives the average execution times in seconds on a laptop of the decision phase (solving of the optimization problem) during one round. They are given for the different approaches and various numbers of sensors. The lack of any optimization explains why the heuristic has very low execution times. Conversely, the strategy with one leader, which requires to solve an optimization problem considering all the nodes presents redhibitory execution times. Moreover,

 ISBN: 978-1-61208-318-6

increasing the network size by 50 nodes multiplies the time by almost a factor of 10. The strategy with two leaders has more suitable times. We think that in distributed fashion the solving of the optimization problem in a subregion can be tackled by sensor nodes. Overall, to be able to deal with very large networks, a distributed method is clearly required.

TABLE I. EXECUTION TIME(S) VS. NUMBER OF SENSORS

Sensors number	Strategy 2 (with two leaders)	Strategy 1 (with one leader)	Simple heuristic
50	0.097	0.189	0.001
100	0.419	1.972	0.0032
150	1.295	13.098	0.0032
200	4.54	169.469	0.0046
250	12.252	1581.163	0.0056

G. The network lifetime

Finally, we have defined the network lifetime as the time until all nodes have been drained of their energy or each sensor network monitoring an area has become disconnected. In Figure 8, the network lifetime for different network sizes and for both strategy with two leaders and the simple heuristic is illustrated. We do not consider anymore the centralized strategy with one leader, because, as shown above, this strategy results in execution times that quickly become unsuitable for a sensor network.

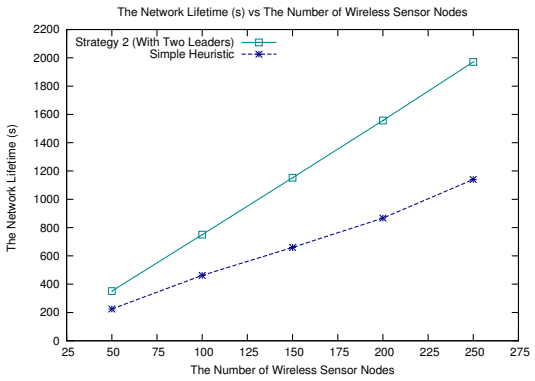


Figure 8. The network lifetime.

As highlighted by Figure 8, the network lifetime obviously increases when the size of the network increases, with our approach that leads to the larger lifetime improvement. By choosing the best suited nodes, for each round, to cover the region of interest and by letting the other ones sleep in order to be used later in next rounds, our strategy efficiently prolonges the network lifetime. Comparison shows that the larger the sensor number is, the more our strategies outperform the simple heuristic. Strategy 2, which uses two leaders, is the best one because it is robust to network disconnection in one subregion. It also means that distributing the algorithm in each node and subdividing the sensing field into many subregions, which are managed independently and simultaneously, is the most relevant way to maximize the lifetime of a network.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have addressed the problem of the coverage and the lifetime optimization in WSNs. To cope with this problem, the field of sensing is divided into smaller subregions using the concept of divide-and-conquer method, and then a multi-rounds coverage protocol will optimize coverage and lifetime performances in each subregion. The proposed protocol combines two efficient techniques: network leader election and sensor activity scheduling, where the challenges include how to select the most efficient leader in each subregion and the best representative active nodes. Results from simulations show the relevance of the proposed protocol in terms of lifetime, coverage ratio, active sensors ratio, energy saving, energy consumption, execution time, and the number of stopped simulation runs due to network disconnection. Indeed, when dealing with large and dense wireless sensor networks, a distributed approach like the one we propose allows to reduce the difficulty of a single global optimization problem by partitioning it in many smaller problems, one per subregion, that can be solved more easily.

In future work, we plan to study a coverage protocol which computes all active sensor schedules in only one step for many rounds, using optimization methods such as swarms optimization or evolutionary algorithms.

REFERENCES

[1] I. F. Akyildiz and M. C. Vuran, Wireless Sensor Networks. John Wiley and Sons Ltd., 2010.

[2] S. Misra, I. Woungang, and S. C. Misra, Guide to Wireless Sensor Networks. Springer-Verlag London Limited, 2009.

[3] S. Misra, M. P. Kumar, and M. S. Obaidat, "Connectivity preserving localized coverage algorithm for area monitoring using wireless sensor networks," Computer Communications, vol. 34, no. 12, 2011, pp. 1484–1496.

[4] A. Nayak and I. Stojmenovic, Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication. John Wiley and Sons, Inc, 2010.

[5] A. Varga, "Omnet++ discrete event simulation system," Available: http://www.omnetpp.org, 2003.

[6] C. T. Vu, "Distributed energy-efficient solutions for area coverage problems in wireless sensor networks," Ph.D. dissertation, Georgia State University, 2009.

[7] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," Mobile Networks and Applications, vol. 10, no. 4, 2005, pp. 519–528.

[8] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Leveraging prediction to improve the coverage of wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 4, 2012, pp. 701–712.

[9] B. Wang, H. B. Lim, and D. Ma, "A coverage-aware clustering protocol for wireless sensor networks," Computer Networks, vol. 56, no. 5, 2012, pp. 1599–1611.

[10] Z. Liu, Q. Zheng, L. Xue, and X. Guan, "A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks," Future Generation Computer Systems, vol. 28, no. 5, 2012, pp. 780–790.

[11] L. Zhang, Q. Zhu, and J. Wang, "Adaptive clustering for maximizing network lifetime and maintaining coverage," Journal of Networks, vol. 8, no. 3, 2013, pp. 616–622.

[12] J. A. Torkestani, "An adaptive energy-efficient area coverage algorithm for wireless sensor networks," Ad Hoc Networks, vol. 11, no. 6, 2013, pp. 1655–1666.

[13] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," Ad Hoc & Sensor Wireless Networks, vol. 1, no. 1-2, 2005, pp. 89–124.

[14] F. Pedraza, A. L. Medaglia, and A. Garcia, "Efficient coverage algorithms for wireless sensor networks," in Proceedings of the 2006 Systems and Information Engineering Design Symposium, 2006, pp. 78–83.

[15] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, vol. 1, no. 4, 2002, pp. 660–670.

 ISBN: 978-1-61208-318-6

MH-LEACH: A Distributed Algorithm for Multi-Hop Communication in Wireless Sensor Networks

José Henrique Brandão Neto, Antoniel da Silva Rego,
André Ribeiro Cardoso, Joaquim Celestino Jr.
Computer Networks and Security Laboratory (LARCES)
State University of Ceará (UECE)
Fortaleza, Brazil
{henrique.brandao, antoniel.rego, andrec, celestino}@larces.uece.br

***Abstract*—Wireless Sensor Networks (WSN) consist in a set of nodes that collect information from the environment and send it to a Base Station that processes the final data. Some challenges in order to minimize the power consumption and maximize the network lifetime in this kind of networks can be found. This paper presents MH-LEACH, an algorithm that permits to establish a multi-hop communication between sensor nodes, which aims to save energy. Using MH-LEACH, a sensor will have options to transmit their data to closer nodes, always sending the collected data to the base station. This proposal was incorporated into the LEACH protocol being evaluated through simulations. The results show improvements in the approach when compared to the original version of LEACH.**

Keywords-Sensor networks; Multi-Hop communication; energy consumption;

I. INTRODUTION

The result of advances in technology and wireless communication, and also sensor networks, has emerged as an important and indispensable tool for the detection of contamination in hazardous environments, habitat monitoring in reserves, enemies inspections in war environments, and other applications [1] [2].

The Wireless Sensor Networks (WSN) are a special kind of ad hoc networks that allow the monitoring of the physical world through small sensors networks densely or sparsely distributed. These networks are composed of hundreds or thousands of sensor nodes with multifunctional low power load, operating autonomously in an environment with limited computational capabilities, and a base station, responsible for receiving data from the sensor nodes.

Currently, WSN are targets of many challenges. One of them is related to the shortage of available energy in sensors, and a large part of the research done today seeks to highlight effective ways to save energy in sensors, making the network lifetime be extended.

The energy used for communication in wireless sensor networks is very high compared to that used for computation, thus it must be carefully used to improve the network lifetime [3]. Routing algorithms based on clustering are widely used to increase the sensor networks lifetime [4] [5] [6].

We present in this article a new algorithm based on clustering that uses a new technique for multihop communication between cluster-heads in order to conserve energy consumed by the network and thereby increase the network lifetime.

This article is organized as follows: Section 2 presents as related work LEACH protocols [7], the LEACH-C [8] and ALEACH [9]. The MH-LEACH is described in Section 3. Section 4 presents the simulations and results. And finally, in Section 5, the conclusion of our work is drawn.

II. RELATED WORK

Saving energy is an extremely important factor in sensor networks. Thus, many routing algorithms aimed in efficient energy consumption have been developed [10] [11] [12]. This section presents some of these algorithms, which are: the LEACH protocol, which is the basis for the development of our work, the LEACH-C and ALEACH.

Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol [7] is a hierarchical protocol for minimizing the power consumption in order to increase the network lifetime. In LEACH, the nodes are organized into clusters, with one node acting as the leader (cluster-head). All non-leaders nodes should transmit their data to the cluster-head, while the cluster-head must receive data from all the cluster members, perform functions of data processing (e.g., data aggregation), and transmit data to the base station.

The LEACH works by rounds. In each round, leader nodes are exchanged in order to distribute the network energy consumption. Two phases compose the rounds: clusters grouping, and communication phase. In the phase of clustering, the choice of leaders is performed through a distributed algorithm, and the source nodes choose to join the nearest cluster-head. In the communication stage, the transfer of data to the base station is made, including aggregation / data fusion by the leaders.

Low-Energy Adaptive Clustering Hierarchy Centralized (LEACH-C) protocol [8] is a variation of LEACH that uses a centralized algorithm for grouping the clusters. During the formation of groups of LEACH-C, each node sends information about its location and energy level to the base station. In order to ensure the distribution of energy to all the nodes in the network, the base station calculates the average energy of the nodes in each round. The cluster-head nodes must have

 ISBN: 978-1-61208-318-6

the energy level above the average to be chosen, and based on them the base station performs the Simulated Annealing algorithm to determine the best cluster-heads. After finding the clusters and the respective leaders, the base station transmits this information to the nodes of the network. The nodes then transmit the data to the leader of their group, which sends data to the base station.

The Advanced Low-Energy Adaptive Clustering Hierarchy (ALEACH) protocol [9] is an efficient energy routing protocol that considers the level of energy in a sensor node in the election of cluster-heads. As in LEACH, the ALEACH works by rounds and does not need to know the geographic positions of the nodes to elect the cluster-heads.

The Multi-hop LEACH protocol [13] represents an extension of the LEACH to save energy in wireless sensor networks. The purpose of the protocol is send the data to the base station through other intermediate nodes. Like LEACH protocol, Multi-hop LEACH uses the same mechanism for the election of clusters-head. At the stage of data collection, two types of communication are allowed: the inter-cluster communication and intra-cluster communication. In the first communication, the network is divided in clusters. The cluster-head of each cluster receives the data of its member nodes. It performs data aggregation and send the final information to the base station through other nodes. In the intra-cluster communication, the members nodes of the cluster send their data to another members nodes to reach the cluster-head. The protocol works by rounds like LEACH and it selects the path with minimum hops between the clusters-head and the base station.

III. The MH-LEACH Algorithm

The main objective of the algorithm is the establishment of multi-hop communication between clusters-head in a network. The main purpose is to send the packet to the nearest cluster-head that is turned towards the Base Station. With this characteristic, it is intended to decrease the power consumption of the nodes and extend the network lifetime, since the smaller the distance to transmit the lower the consumption is.

A. Energy Consumption Model

As described in [8], a sensor node spends energy according to the model shown in (1) and (3).

The transmission of a message of k bits at a distance d has the following energy consumption:

$$E_{Tx}(k,d) = E_{Tx-elec}(k) + E_{Tx-amp}(k,d) \quad (1)$$

$$E_{Tx}(k,d) = \begin{cases} E_{elec} * k + \epsilon_{fs} * k * d^2, & d < d_0 \\ E_{elec} * k + \epsilon_{mp} * k * d^4, & d \geq d_0 \end{cases} \quad (2)$$

and receiving a message, the sensor has the following consumption:

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} * k, \quad (3)$$

where:

$E_{Tx-elec}$ = Energy spent in transmission;
$E_{Rx-elec}$ = Energy spent in receiving data;
E_{Tx-amp} = Energy of transmission amplifier;
d_0 = threshold distance, calculated according to the values of E_{elec}, ϵ_{fs} e ϵ_{mp};
ϵ_{fs} = Parameter called free space model (fs), is used if the distance from source to target is less than d_0;
ϵ_{mp} = Parameter called multipath model (mp), used if the distance from source to target is greater than or equal to d_0;
E_{elec} = Energy spent per bit transmitted or received

B. MH-LEACH Operation

In order to understand the proposition, it is important to point out that during the construction of routes between the transmitter (cluster-head) and base station, it is assumed that the network is already clustered and the cluster-heads of each group are already set.

One of the main goals of the algorithm is to find possible routes for a cluster-head (leader) to send a packet using other cluster-heads in order to save its energy. The choice of the next cluster head to get the message must take in account if it has enough energy. Thus, if a cluster-head cannot send a message for another one, this node will try to find another cluster-head based on information contained in its routing table, according to described ahead.

This proposal takes into account the fact that the higher the signal strength of the received packet, i.e., Received Signal Strength Indicator (RSSI), the greater the proximity of the node that sent the message. This information is used in order to build the routing table for each cluster-head.

The MH-LEACH proposes the routes establishment using two phases:

- ***Phase 1:*** The cluster-heads are defined as a part of LEACH algorithm. After that, they broadcast an announcement message and all the cluster-headers in the transmission ratio will take the advantage in order construct their routing table taking in account the level of signal (RSSI) received. So, they organize their early routes containing the closest cluster-heads to send a packet. The base station performs the same procedure as seen in Fig. 1.
- ***Phase 2:*** After that, each leader sends these initial routes (from routing table) to the base station that will check whether a cluster-head can be in the route of another one. After this check, the base station sends their routes back to the nodes.

This procedure is necessary because it needs to create a table of possible routes to a cluster-head. From the intensity of the signal announced, each node keeps a list sorted by proximity of the possible destinations of the packet. As shown in Fig. 2, node 1 has the first choice route the node 2, node 2 to the node 3, and so on. The id (identifier) zero in the table indicates the Base Station. Negative values indicate the signal strength in decibelmiliwatt (dBm).

 ISBN: 978-1-61208-318-6

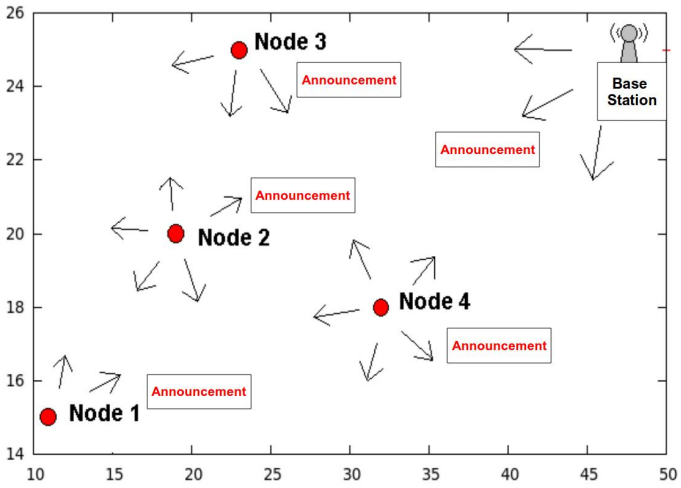


Figure. 1. Cluster-head and base station are identified.

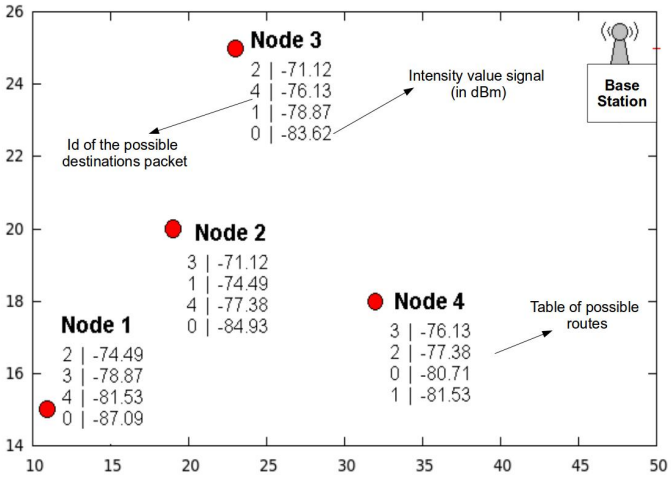


Figure. 2. Creation of each node initial tables.

Despite this initial table indicates the nearest cluster-head node to pass the packet, it contains wrong routes. For example, node 2 has as first route the node 3, but this node has node 2 as well as the first route, causing a loop in the network. Another wrong fact is that, node 2 has node 1 as route option, but it is not a good alternative because the packet is transferred in an opposite direction to the Base Station.

To solve these problems, phase 2 of the algorithm is performed. Each cluster-head sends their initial table to the base station so that it can check and correct them.

When the base station obtains all tables from all cluster-head, it performs an algorithm to determine whether a cluster-head can be in another cluster-head′s route table. The algorithm can be seen below:

$$IF\ \{\ I-noX-EB < I-noY-EB\ \}\ THEN$$

"The Y node is a possible route to the X node"

where:

$I-noX-EB$ = Intensity of a packet that the node X received from base Station

$I-noX-EB$ = Intensity of a packet that the node Y received from base Station

If the node Y received a packet of the base station with an higher RSSI than node X, it means node Y is closer to the base station than node X. The figures below show the performance of this algorithm to correct the tables assembled in Phase 1.

As seen in Fig. 3, the Base Station corrects the initial routing table of the node 2. It checks whether the node 3 may be a possible route to node 2. Since the test is satisfied, the node 3 remains in the table.

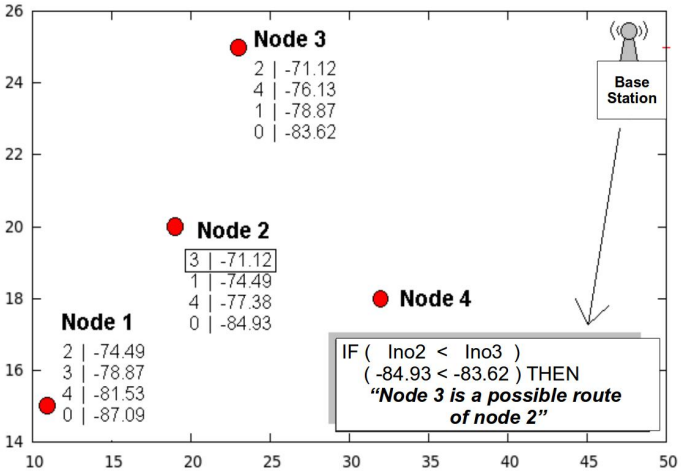


Figure. 3. Checking if node 3 can be a route to node 2.

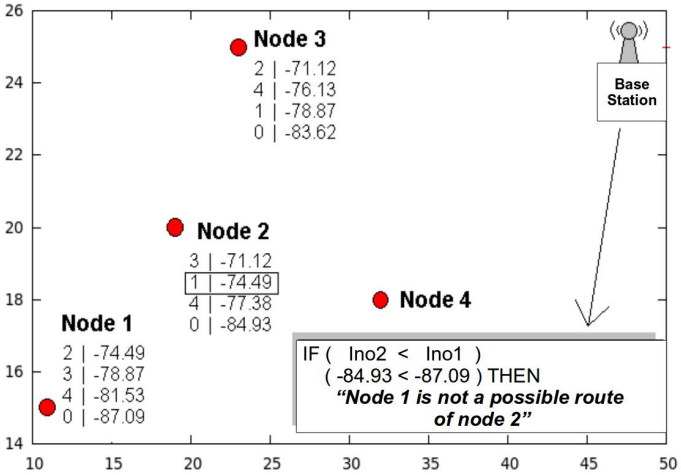


Figure. 4. Checking if node 1 can be a route to node 2.

The test done in node 1 is not satisfied, then it will not be part of the routing table of node 2. The procedure can be seen in Fig. 4.

Node 4 is approved and remains as a route of node 2 as shown in Fig. 5. The Base Station indicated by Id (identifier) zero in the table always remains; it is one last option route of

 ISBN: 978-1-61208-318-6

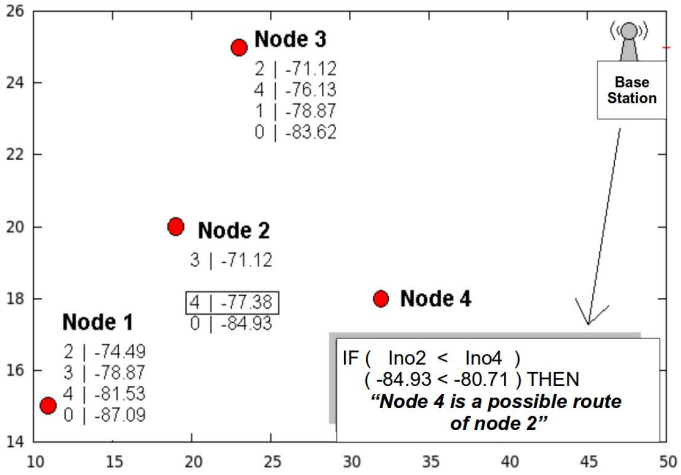


Figure. 5. Checking if node 4 can be a route to node 2.

the node. The reason is, if it is not possible to send data to any node in the network, it sends it to the base station.

After reviewing the table of each cluster-head, the base station sends back to the nodes the correct tables, free of loops and wrong forwards that aren't in its direction. Fig. 6 shows the final result table of each node after the checking performed by the base station.

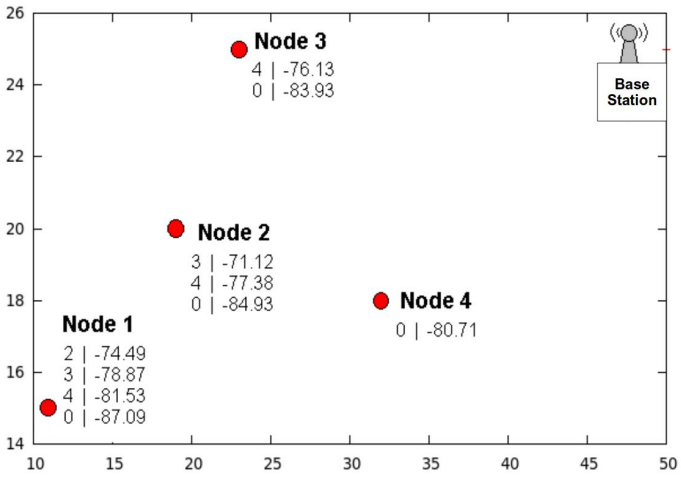


Figure. 6. Final result of route tables.

Thus, the process of calculating routes ends, where each leader node will not transmit directly to the base station, but for some other closer cluster-head to it indicated in its routing table. After that, the collection and transfer of data is started.

IV. Results

We evaluated the performance and validity of the proposed algorithm through simulations. The proposal was incorporated into the LEACH protocol making it multi-hop. We compared the obtained results of this approach to the results of the original version of LEACH using Castalia simulator [14]. This simulator is designed specifically for sensor networks being an extended module of OMNET++ simulator. Castalia has realistic wireless modeling channel and radio modules, where nodes behave very close to reality in the use of the radio.

The metrics used in the simulation were:

- ***Average consumption per node.*** In this metric, it is evaluated if the algorithm provided energy savings to the sensor nodes.
- ***Sending cost per energy consumed.*** This metric represents the total number of packets transmitted by the average consumption of each node. It indicates that the network took it best energy capacity due to the multi-hop calculation. It also indicates that the network has consumed less energy and sent a greater amount of packets.
- ***Time of death of the first node.*** This metric checks the network coverage time, i.e., the time that the network works 100% in data collection. Since a node dies, that coverage is no longer full.
- ***Time of Death 80% of nodes.*** With this metric, it is possible to check how long the network survived. Thus, we can notice whether the proposed idea has extended the network life.

The scenarios used in the simulation and evaluation of results are defined in Table 1. The energy of each node was set to 5 joules. The simulation time was 60 seconds. This period of time was set because at the end of that value most of nodes are almost inactive since the initial energy is low. The nodes were randomly distributed in the created area. The time for each round of the LEACH protocol was 20s. The number of cluster-head in every round was 5. Each simulation was run 33 times defining a confidence interval of 95% for all results.

TABLE I
Scenarios used in simulations

Scenario	Number of nodes	Area (m x m)	Base Station Position
1	50	50 X 50	(25, 100)
2	100	70 X 70	(35, 140)

Figures 7 and 8 show the results related to the average energy consumption of the nodes in the two scenarios, respectively.

It is possible to see in the first scenario that in the MH-LEACH protocol has better average power consumption when compared to LEACH. The values of the average consumption were 1.82097 and 1.63964 for LEACH and MH-LEACH respectively. The gain of the proposed approach was approximately 9%.

In the second scenario, the MH-LEACH protocol also obtained better values compared to the original LEACH. The results for the average consumption were 1.69509 and 1.59406 for LEACH and MH-LEACH respectively. The gain was approximately 6%.

Figures 9 and 10 show the results for the total number of packets transmitted by the average energy consumed. The higher that value, the better the protocol performance, since it was possible to send more packets with a low consumption.

 ISBN: 978-1-61208-318-6

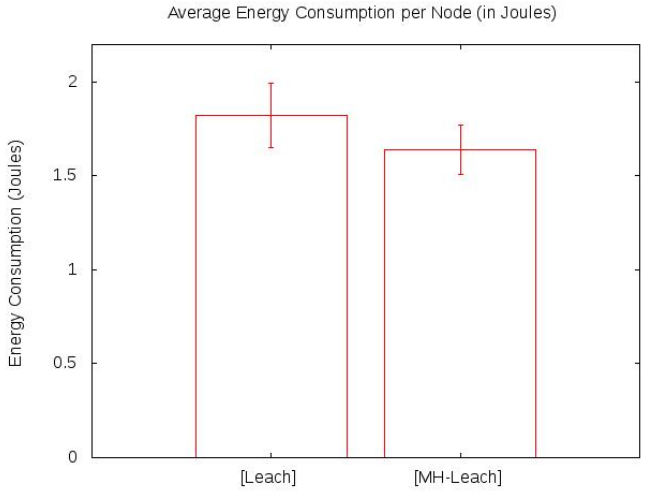


Figure. 7. Average energy consumption per node Scenario 1.

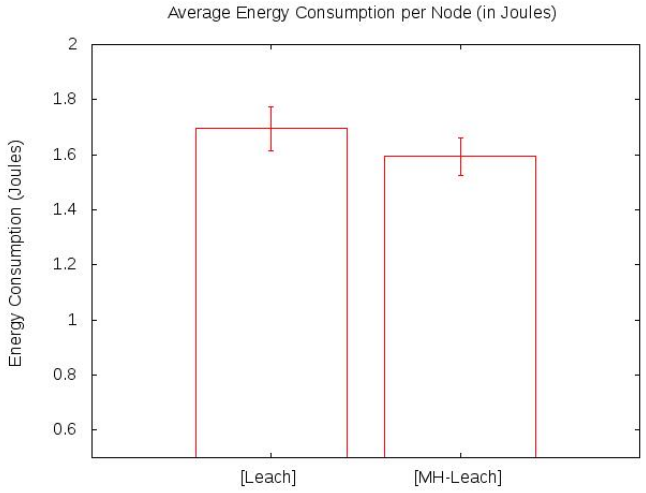


Figure. 8. Average energy consumption per node Scenario 2

In scenario 1, the LEACH and MH-LEACH obtained the following results, 99.765 and 129.723 respectively.

In scenario 2, the results were 236.798 and 266.671 for LEACH and MH-LEACH respectively. As can be seen, the LEACH protocol using the new algorithm achieved a better rate in both scenarios, i.e., more packets are sent with an energy efficient consumption.

Figures 11 and 12 show the results for the metric related for the first time that the network node dies. In scenario 1, the times of death were 22.4115 and 23.4351 for LEACH and MH-LEACH respectively. The results show that the proposed approach had a greater coverage time.

In scenario 2, the results showed the following values, 22.1991 and 23.2233 for LEACH and MH-LEACH respectively. In this scenario, the LEACH protocol with the proposed algorithm once again obtained better results than the original version of the protocol.

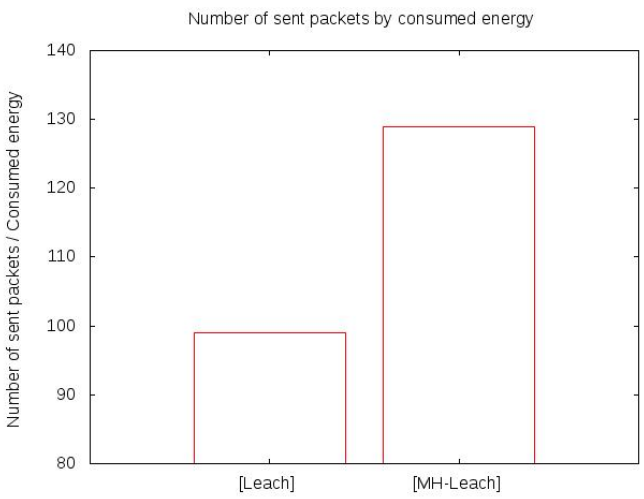


Figure. 9. Number of sent packets by consumed energy Scenario 1

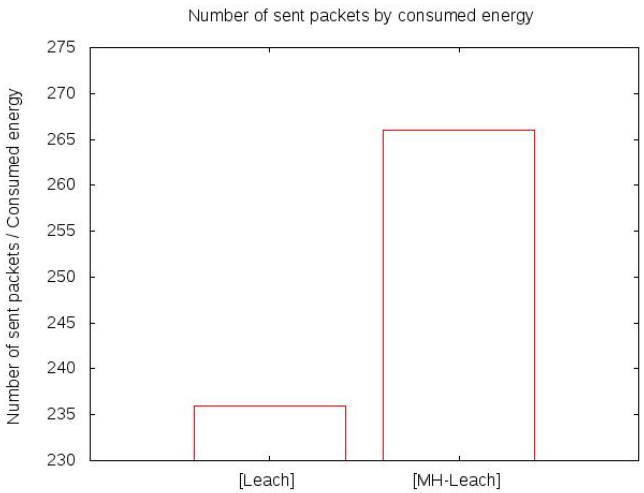


Figure. 10. Number of sent packets by consumed energy Scenario 2

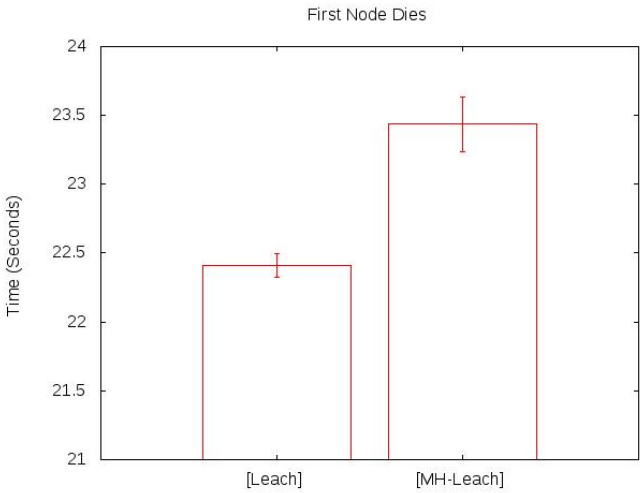


Figure. 11. Time of first node death Scenario 1

 ISBN: 978-1-61208-318-6

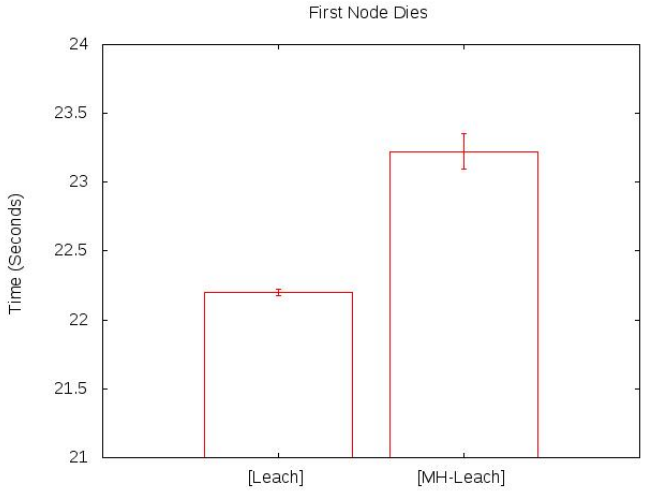


Figure. 12. Time of first node death Scenario 2

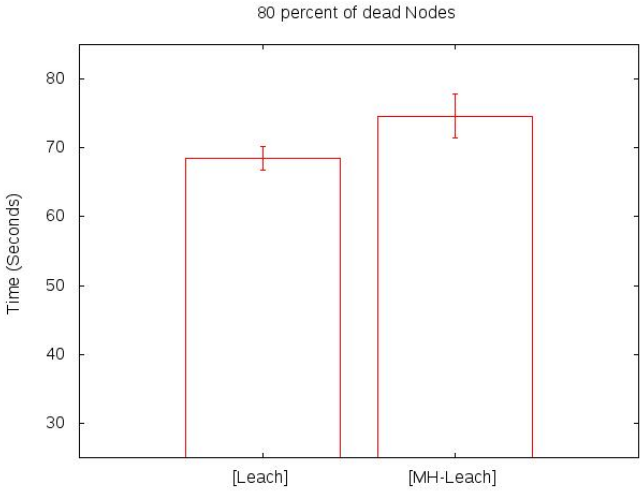


Figure. 14. Time for 80 percent of dead nodes Scenario 2

Figures 13 and 14 show the results for the metric that shows the time of death of 80% of nodes. There is a small detail concerning the calculation in this metric. The simulation time was increased to 120s generating a new simulation. This extension was done to ensure the collection of the times of death of at least 80% of nodes.

The times results obtained in scenario 1 were 54.1589 and 64.1114 for LEACH and MH-LEACH respectively. The new algorithm has obtained a longer time indicating that the network survived longer, i.e., since the nodes saved more energy transmitting to closer nodes, the network lifetime was extended.

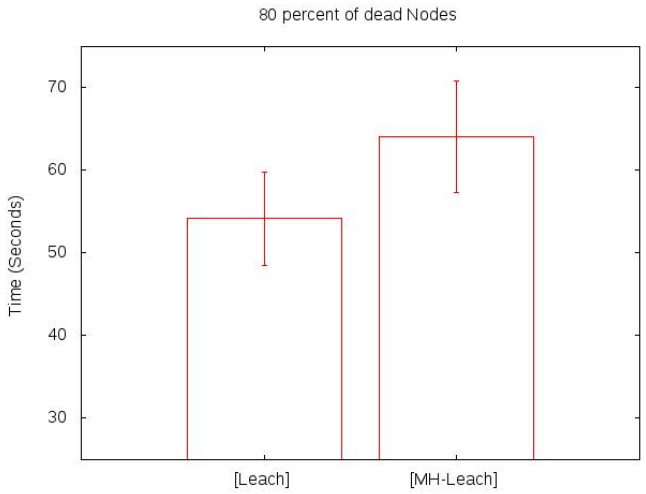


Figure. 13. Time for 80 percent of dead nodes Scenario 1

In scenario 2, concerning this metric, the following values were found: 68.4805 and for LEACH and 74.6026 for MH-LEACH. It is observed that the proposed idea got better results when compared to other protocol.

V. Conclusion and Future Work

The purpose of the study was to develop an algorithm for establishing multi-hop communication between sensor nodes in a network, with the main goal of saving energy. To achieve this purpose, we used the feature that greater signal intensities indicate node proximity. The base station was also used in order to perform a centered calculation to avoid errors on the use of found routes.

From the experiments, it was observed that the new algorithm achieved improvements when compared to the LEACH protocol. The gains were observed in power consumption and network lifetime, which was extended.

We conclude that the proposal is presented as an interesting idea that saves energy in sensor networks, which can be adapted to other single-hop protocols to achieve improvements in their running.

For future works, the proposal will be to adapt this technique in others protocols, like LEACH-C and ALEACH. Another future activity, it will be the development of a mechanism for the cluster-head node to use other possible routes of its table within a round taking into account the battery remains in the neighboring leaders.

References

[1] Q. Jiang and D. Manivannan, “Routing protocols for sensor networks,” Proc. Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE. IEEE, 2004, pp. 93–98.

[2] S. D. Muruganathan, D. C. Ma, R. I. Bhasin, and A. O. Fapojuwo, “A centralized energy-efficient routing protocol for wireless sensor networks,” Communications Magazine, IEEE, vol. 43, no. 3, 2005, pp. S8–13.

[3] A. Thakkar and K. Kotecha, “Wcvaleach: Weight and coverage based energy efficient advanced leach,” vol. 2, no. 6, pp. 51–54, 2012.

[4] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” Computer networks, vol. 51, no. 4, 2007, pp. 921–960.

[5] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” Wireless Communications, IEEE, vol. 11, no. 6, 2004, pp. 6–28.

 ISBN: 978-1-61208-318-6

[6] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” Ad hoc networks, vol. 3, no. 3, 2005, pp. 325–349.
[7] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” Proc. System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on. IEEE, 2000, pp. 10–pp.
[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” Wireless Communications, IEEE Transactions on, vol. 1, no. 4, October 2002, pp. 660–670.
[9] M. S. Ali, T. Dey, and R. Biswas, “Aleach: Advanced leach routing protocol for wireless microsensor networks,” Proc. Electrical and Computer Engineering, 2008. ICECE 2008. International Conference on. IEEE, December 2008, pp. 909–914.
[10] D. Kandris, P. Tsioumas, A. Tzes, G. Nikolakopoulos, and D. D. Vergados, “Power conservation through energy efficient routing in wireless sensor networks,” Sensors, vol. 9, no. 9, September 2009, pp. 7320–7342.
[11] K. T. Kim and H. Y. Youn, “Energy-driven adaptive clustering hierarchy (edach) for wireless sensor networks,” Proc. Embedded and Ubiquitous Computing–EUC 2005 Workshops. Springer, 2005, pp. 1098–1107.
[12] M. Liu, J. Cao, G. Chen, and X. Wang, “An energy-aware routing protocol in wireless sensor networks,” Sensors, vol. 9, no. 1, January 2009, pp. 445–462.
[13] R. V. Biradar, D. Sawant, D. Mudholkar, and D. Patil, “Multi-hop routing in self-organizing wireless sensor networks,” IJCSI International Journal of Computer Science, vol. 8, no. 1, January 2011, pp. 154–164.
[14] Castalia simulator. [Online]. Available: http://castalia.npc.nicta.com.au, Retrieved: December, 2013.

 ISBN: 978-1-61208-318-6

QoE-based Adaptive mVoIP Service Architecture in SDN Networks

Dongwoo Kwon, Rottanakvong Thay, Hyeonwoo Kim, and Hongtaek Ju
Department of Computer Engineering
Keimyung University
Daegu, Republic of Korea
e-mail: {dwkwon, vongjacky91, hwkim84, juht}@kmu.ac.kr

***Abstract*—In this paper, we propose the adaptive Mobile Voice over Internet Protocol (mVoIP) service architecture in Software Defined Networking (SDN) networks to provide the best quality of mVoIP service to end-users. The key challenges in improving the mVoIP Quality of Experience (QoE) are forwarding data path optimization for VoIP flows and optimized codec selection with consideration of network congestion. Based on network Quality of Service (QoS) data and predicted mVoIP QoE data to be collected in SDN networks through mVoIP agents, the proposed service architecture improves mVoIP QoS. In particular, this paper focuses on the improvement of mVoIP QoS by adaptive codec selection optimization and proposes an algorithm for adaptive codec selection in SDN networks.**

Keywords-VoIP; mVoIP QoE; Adaptive mVoIP service; Codec selection optimization; Network QoS; SDN

I. INTRODUCTION

Voice over Internet Protocol (VoIP) [1] has become one of the most widely used protocols for voice service delivery. It has been used not only for public services, such as Skype and Viber, but also for private services for companies and organizations. The quality of a VoIP service mostly depends on the voice codec, which is determined by network capability and the number of users to be accommodated. Thus, to improve the quality of a VoIP service, two factors should be considered: VoIP flow control and optimized codec selection.

In Software Defined Networking (SDN) [2] networks, which have provided advances in network management, VoIP flows can be controlled by an OpenFlow [2] controller, such as NOX/POX, Floodlight, Ryu, or OpenDaylight, to guarantee Quality of Service (QoS). To adjust for network congestion, flow tables of OpenFlow switches need to be managed by optimized forwarding path decisions.

Codec selection is another important factor. Due to variation in the number of active users, a fixed voice codec negotiated between a VoIP server and VoIP clients without consideration of the network situation will cause inefficiency and unavailability. In particular, if using a low-quality codec under conditions of spare bandwidth, the utilization of network bandwidth is inefficient; if using a high-quality codec under conditions of insufficient bandwidth, the network bandwidth becomes unavailable. To overcome these limitations, a method of network situation-aware codec selection [3]-[7] is required.

In this paper, we propose the adaptive Mobile VoIP (mVoIP) service architecture based on network QoS data and predicted mVoIP Quality of Experience (QoE) data in SDN networks. The design of our service architecture was guided by two principles: (1) Keep it simple, able to be applied without any modification of VoIP client applications, with an IP Private Branch Exchange (PBX) system, as a practice for VoIP services, (2) Satisfy the Service Level Agreement (SLA) for the VoIP service.

In the proposed architecture, mVoIP QoS is guaranteed by forwarding data path control for VoIP flows and optimized codec selection with consideration of network congestion. To gather QoS data in a wireless network and predict the mVoIP QoE, mVoIP QoE measurement agents are deployed in the wireless network. An mVoIP QoS manager, as a SDN application, collects QoS data on the SDN network from OpenFlow switches and mVoIP QoE data from mVoIP QoE measurement agents. Then, it decides whether the forwarding path for mVoIP flows and the voice codec are optimal under specific network conditions. A selected voice codec is applied by an mVoIP QoS adapting agent within an IP PBX system in the SDN network.

This paper focuses on adaptive codec selection followed by the network QoS and mVoIP QoE data, while the forwarding path decision in SDN networks depends on an OpenFlow controller and a network slicer such as FlowVisor [8]. The rest of this paper is organized as follows. Related studies are discussed in Section II. In Section III, the adaptive mVoIP service architecture in SDN networks is described in detail, including the mVoIP QoS manager, the mVoIP QoS adapting agent, and the mVoIP QoE measurement agent. The algorithm for adaptive codec selection in SDN networks is also proposed. In Section IV, the conclusions are presented and future works are discussed.

II. RELATED WORK

Takahashi et al. [9] presented the factors that determine QoS for a VoIP service and a subjective/objective quality assessment. For subjective quality assessment, opinion rating and the opinion equivalent-Q method are introduced. Opinion models such as E-Model and speech-layer objective models such as Perceptual Evaluation of Speech Quality (PESQ), P.563, and P.AAM are introduced for objective quality assessment. Packet-layer objective models such as P.VTQ are also introduced for objective quality assessment, which is based merely on IP packet information except for

 ISBN: 978-1-61208-318-6

speech data in the payload. To investigate the relationship between subjective and estimated quality, the experiment using the modified E-Model was conducted, and the results were presented. Finally, a framework for the development of quality assessment research was proposed.

Kim et al. [10] measured PESQ scores to compare the performance of various codecs, such as G.711, GSM, iLBC, Speex, and Skype's codec, in wireless networks. The experiments were performed by transmitting audio signals from one VoIP client to another using the Jack audio router through the NIST Net emulator to measure the performance of the VoIP codecs. As a result of these experiments, these researchers found that Skype was the most robust against packet delay and packet loss. Among codecs other than commercial services, Speex was robust when packet loss was less than 10% and iLBC was very robust under network congestion. These results can be utilized to initialize default codec sets for the adaptive codec selection method that we propose in this paper.

Sfairopoulou et al. [3] evaluated the performance of a combined Call Admission Control with Codec Selection (CAC/CS) algorithm proposed in [4], based on three policies: non-adaptive policies that start with the lowest and highest bandwidth codecs, and an adaptive policy that changes codec in randomly active calls. In particular, experiments were conducted applying the adaptive policy to three cases: applying codec selection (1) only to new call requests (*on new*), (2) only in the presence of rate changes (*on rate*), or (3) both for new calls and at any rate change (*on both*). The o*n new* case presented a low blocking probability but high dropping probability; the *on rate* case maintained high estimated Mean Opinion Score (MOS) values under high traffic load; and the *on both* case presented a trade-off between blocking and dropping probabilities but the worst estimated MOS values. In our study, the proposed architecture involves controlling VoIP flow paths according to network congestion to guarantee a predefined Call Setup Success Rate (CSSR) and Call Drop Rate (CDR), comparing them with the CSSR and CDR obtained from an IP PBX system. Also, the codec selection method considers the number of active calls and the movement of each VoIP client.

Ng et al. [5] proposed and evaluated an adaptive codec switching algorithm for VoIP applications in wired and wireless networks. Codec switching during active calling is achieved by renegotiating the audio session using a RE-INVITE message in Session Description Protocol (SDP). To evaluate the proposed algorithm, a Session Initiation Protocol (SIP) proxy server was deployed on the real network and SIP clients were connected to the wireless Access Point (AP). To simulate network congestion, the network traffic emulator was also deployed on the wireless network. The result was effective in increasing the average MOS values when switching between PCMU and GSM, while the packet loss rate was more than 16%. However, only the packet loss rate was considered to adaptively determine the optimized codec in this study. To apply the proposed algorithm, SIP client applications should be modified based on the calculated packet loss rate. In our proposed architecture, both a client application and an IP PBX system require no modification to be practical.

Roychoudhuri et al. [6] proposed adaptive rate control for audio packets, which is based on packet loss prediction and on-line audio quality assessment. Audio Genome was used to store the codec type, loss distribution, and delay to derive the audio quality of the ongoing transmission. The adaptive rate control framework was proposed to combine all audio codecs with Audio Genome to maintain optimized audio quality. In the proposed framework, the Rate-Quality Optimization problem is to maximize the audio quality under the constraints of available bandwidth and delay. The experiment for Rate-Quality Optimization was conducted with the codecs PCMU, G.721, GSM FR, G.728, and G.723.1 in six scenarios. In the scenario of low available bandwidth and low delay among them, the feasible solution was G.728 (61%) and PCMU (39%). In the worst-case scenario of low available bandwidth and high delay, the feasible solution was G.721 (72%), G.728 (19%), and PCMU (8%). One of our approaches is to determine the priority set of optimized codecs and quality options, when VoIP client applications and an IP PBX system are used without any modification. Thus, these results can be used to create default codec sets in our adaptive codec selection method with [10].

Qiao et al. [7] proposed a QoS control scheme that combines rate-adaptive and priority marking QoS control to improve speech quality. A VoIP simulation system with a NS-2 network simulator was used to simulate VoIP flow, which includes an encoder/decoder/marker for the Adaptive Multi-Rate (AMR) codec, a bitrate controller, and a loss simulator. The MOS scores were predicted using PESQ based on given AMR rates and packet loss. Because each VoIP client application supports different codecs, a set of diverse codecs including non-multi-rate codecs is applied in our adaptive codec selection method.

III. Adaptive mVoIP Service Architecture

The quality of VoIP service depends on the codec used and the network condition. If high-quality codecs are used for the VoIP service, the voice quality is very good, but the number of VoIP channels is less than in the case of using low-quality codecs due to VoIP bandwidth consumption.

The bandwidth consumption, which is represented by *B*, can be calculated by (1). The codec bit rate and packets per second (PPS) are calculated by dividing the codec sample size (*SS*) by the codec sample interval (*SI*) and dividing the codec bit rate by the voice payload size (*PS*), respectively. The total bandwidth consumption for the VoIP service can be calculated by multiplying the VoIP packet size (*P*) by PPS and the number of channels (*C*), while *P* is represented by the sum of data link layer header, the IP/UDP/RTP header, and the voice payload size.

$$B = (SS \times P \times C) / (SI \times PS) \quad (1)$$

In SDN networks, OpenFlow switches maintain their flow tables based on the FlowModify messages which are sent from the OpenFlow controller to determine the shortest

 ISBN: 978-1-61208-318-6

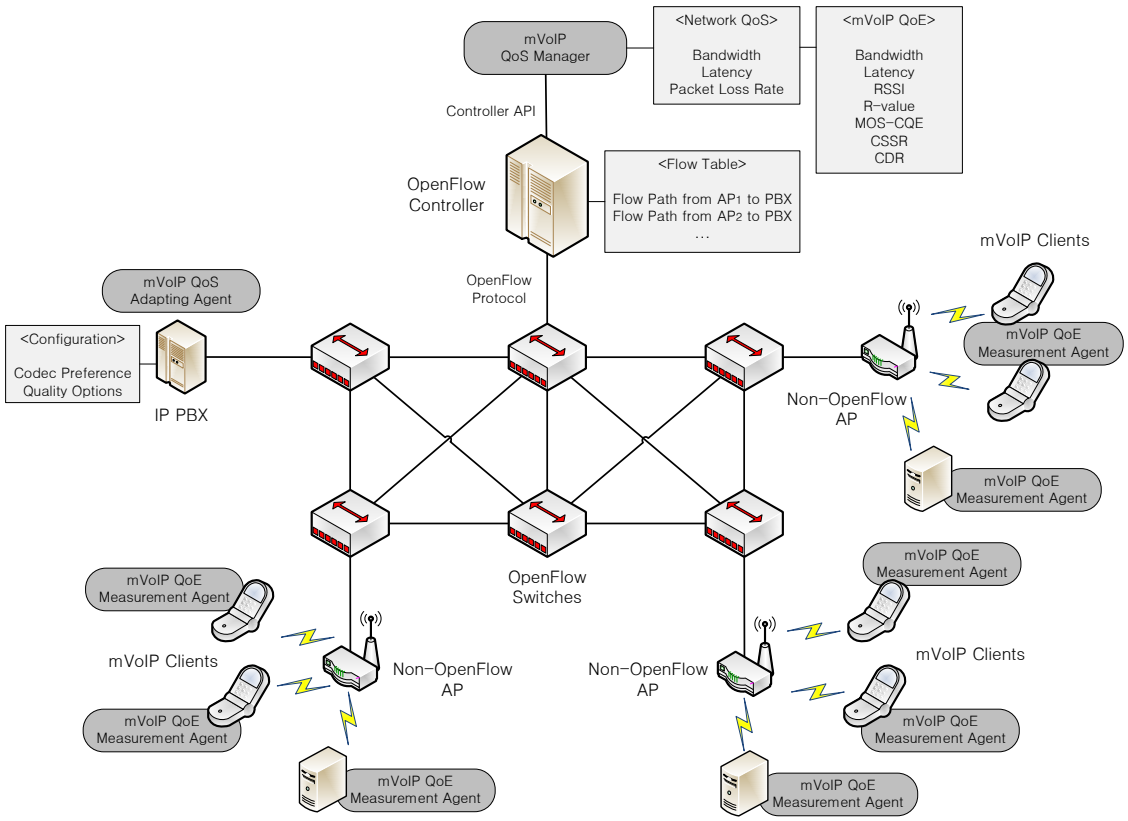


Figure 1. Adaptive mVoIP Service Architecture in SDN Networks

packet paths. To improve and guarantee VoIP QoS, the flow paths between the IP PBX system and the VoIP clients should be determined by consideration of the network conditions, such as bandwidth and latency. After that, the optimized codec can be selected based on the bandwidth usage rate and the channel requirement. It allows the priority of the codec to be rearranged based on the network condition and the number of currently active users of the VoIP service.

This paper proposes an adaptive mVoIP service architecture in SDN networks, which consists of the mVoIP QoS manager, mVoIP QoE measurement agents, and the mVoIP QoS adapting agent as shown in Fig. 1. The SDN network connects with an OpenFlow controller, OpenFlow switches, Non-OpenFlow wireless APs for connecting to mVoIP clients, and an IP PBX system.

A. mVoIP QoE Measurement Agent

In the proposed architecture, mVoIP QoE measurement agents on the dedicated systems conduct bandwidth and latency measurements from themselves to the IP PBX system. Although the controller has all the information on the network topology and can measure bandwidth and latency by itself, the agents need to measure them because non-OpenFlow APs are regarded as edge nodes in SDN networks, and the links between an AP and its connected devices may be critical bottlenecks. These measurements are not conducted by the agents on mobile devices due to excessive battery consumption.

The agent on a mobile device merely collects the Received Signal Strength Indication (RSSI) value from broadcasting information, sent by an AP, to judge the mobility of the VoIP clients. The movement of the VoIP client can be judged by GPS, gyroscope, and RSSI data. Because the GPS or the gyroscope sensor of a mobile device deals with movement in very wide or narrow areas, respectively, the RSSI value is used to judge the mobility of the device. It also directly reflects the wireless network condition. Thus, frequent variation of the RSSI value shows that a VoIP client is moving and implies that a codec with low bandwidth consumption is determined in spite of sufficient available bandwidth.

The VoIP quality metrics are categorized as call setup quality and call quality. Call setup quality is estimated by CSSR and CDR, which can be gathered from the mVoIP QoS adapting agent that communicates with the IP PBX system. Low CSSR and CDR show that the flow path between the AP and the IP PBX system should be modified to avoid network congestion.

Call quality can be measured by a subjective MOS test. Due to time-consuming efforts and expensive cost, the E-Model is used as an alternative to human-based MOS estimation to predict call quality. The gross score, which is represented by the *R*-value, of the E-Model is computed by:

$$R = Ro - Is - Id - Ie + A \quad (2)$$

 ISBN: 978-1-61208-318-6

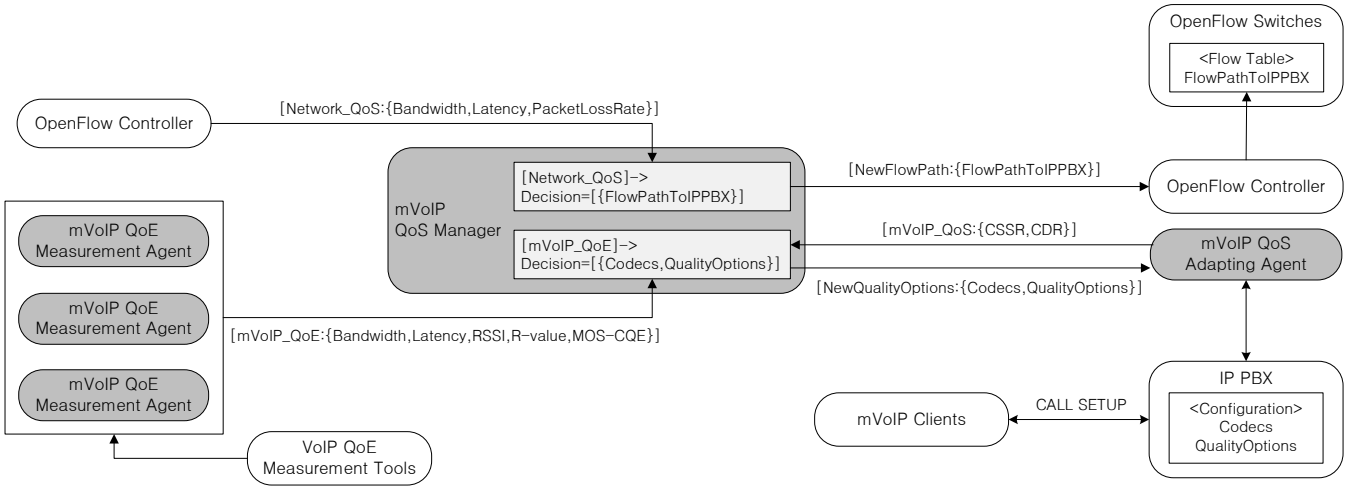


Figure 2. mVoIP QoS Manager

where *Ro* represents the basic signal-to-noise ratio; *Is* represents all impairments that occur simultaneously with the voice signal; *Id* represents all impairments caused by delay and echo effects; *Ie* represents impairments caused by low bit-rate codecs; and *A* is an advantage factor that allows for an advantage of access. *R*-value can be translated to MOS-Conversational Quality Estimated (MOS-CQE) by (3) [11],[12].

$$MOS\text{-}CQE = \begin{cases} 1 & (R<0) \\ 1 + 0.035R + R(R-60)(100-R)\cdot 7\cdot 10^{-6} & (0<R<100) \\ 4.5 & (R>100) \end{cases} \quad (3)$$

R-value is calculated by using VoIP QoE measurement tools, such as VoIPmonitor [13] and pjsip-perf [14], and is transformed into a MOS scale by the agents. Then, the agents send all the gathered data to the mVoIP QoS manager, including the bandwidth, the latency, and the RSSI data, to adaptively select the optimized codec.

B. mVoIP QoS Manager

Based on the network QoS and estimated mVoIP QoE information, such as MOS-CQE, the mVoIP QoS manager determines the flow path between an AP and the IP PBX system and chooses the codec preference. The new flow path is updated by the OpenFlow controller, and the new codec preference is set by the mVoIP QoS adapting agent into the IP PBX system. Fig. 2 depicts data flow among the manager and agents on the proposed architecture.

The algorithm for adaptive codec selection according to network conditions and VoIP service policies is described in Table I. First of all, before determining an optimized codec preference set, the flow paths among the APs and the IP PBX system are recalculated, particularly when CSSR or CDR does not satisfy the predefined threshold. For example, in Korea, the CSSR and CDR thresholds can be defined as 95% and 5%, respectively, according to the standard proposed by Telecommunication Technology Association (TTA) [15].

In the proposed algorithm, the codec set consuming minimal bandwidth is selected when the number of current active users is greater than the designed channel requirement, the network latency is lower than the predefined latency threshold, or the mVoIP client is moving. A latency threshold can be defined at 150ms, as proposed by TTA [15].

For premium users, the codecs providing the best voice quality, such as PCM, under the available bandwidth are selected when not group-calling. For the other users, the codec with the highest MOS-CQE in the minimal bandwidth set among the default sets of the APs is set if the consumed bandwidth of the selected codec is under the available bandwidth and the average bandwidth-consuming current available channel with extra one channel is also under the available bandwidth.

TABLE I. PROPOSED ADAPTIVE CODEC SELECTION METHOD

Algorithm: Adaptive Codec Selection for mVoIP in SDN Networks	
input	Caller: Cr Set of Callees: $CE = \{ CE_1, CE_2, \ldots, CE_n \}$ AP set: $AP = \{ AP_1, AP_2, \ldots, AP_n \}$ Available bandwidth set of each AP: $B = \{ B_1, B_2, \ldots, B_n \}$ Latency set of each AP: $L = \{ L_1, L_2, \ldots, L_n \}$ Set of default codec set of each AP according to channel requirements: $DEFset = \{ DEFset_1, DEFset_2, \ldots, DEFset_n \}$ Set of average variation per unit time of RSSI of each client connected to the specific AP: $R = \{ R_1, R_2, \ldots, R_n \}$ MOS-CQE set of each AP: $MOS = \{ MOS_1, MOS_2, \ldots, MOS_n \}$ CSSR set of each AP: $CSSR = \{ CSSR_1, CSSR_2, \ldots, CSSR_n \}$ CDR set of each AP: $CDR = \{ CDR_1, CDR_2, \ldots, CDR_n \}$ Supported codec set (sorted by low bandwidth and high packet interval): $C = \{ C_1, C_2, \ldots, C_n \}$ Required channel number set of each AP: $REQ = \{ REQ_1, REQ_2, \ldots, REQ_n \}$ Active user set in IP PBX: $USER = \{ USER_1, \ldots, USER_n \}$ Premium user set: $PUSER = \{ PUSER_1, PUSER_2, \ldots, PUSER_n \}$ Predefined CSSR threshold set of each AP: $CSSRTHR = \{ CSSRTHR_1, CSSRTHR_2, \ldots, CSSRTHR_n \}$ Predefined CDR threshold set of each AP: $CDRTHR = \{ CDRTHR_1, CDRTHR_2, \ldots, CDRTHR_n \}$ Predefined latency threshold set of each AP: $LTTHR = \{ LTTHR_1, LTTHR_2, \ldots, LTTHR_n \}$ Predefined RSSI variation value to infer whether a client is moving: $MRssi$

output	Priority set of selected codecs: $S = \{ S_1, S_2, \ldots, S_n \}$ Quality option set of selected codecs: $Q = \{ Q_1, Q_2, \ldots, Q_n \}$
1	$S \leftarrow \{ \}$
2	$Q \leftarrow \{ \}$
3	$apcr \leftarrow AP_{i \supseteq \{Cr\}}$
4	$apce \leftarrow \{ AP_{i \supseteq \{CE1\}}, AP_{j \supseteq \{CE2\}}, \ldots, AP_{k \supseteq \{CEn\}} \}$
5	$ap \leftarrow apcr \cup apce$
6	**if** $CSSR_{ap} < CSSRTHR_{ap}$ **or** $CDR_{ap} > CDRTHR_{ap}$:
7	Request to the controller to assign the optimized flow path between the APs, which have low CSSR or high CDR, and the IP PBX system
8	**if** $NewFlowPath_{(NODEap \text{ to } NODEpbx)} = PreviousFlowPath_{(NODEap \text{ to } NODEpbx)}$:
9	$DEF_{min} \leftarrow \min_{bandwidth}(DEFset_{ap})$
10	S $\leftarrow DEF_{min}$
11	$Q \leftarrow DEF_{min}.option$
12	**return**
13	**end if**
14	**end if**
15	**if** $NUM_{USERi \in ap} > REQ_{ap}$ **or** $L_{ap} > LTTHR_{ap}$ **or** ($apcr_{RCr} \geq MRssi$ **or** $apce_{RCEi} \geq MRssi$):
16	$C_i \leftarrow \max_{MOS\text{-}CQE}(C_1, \max_{MOS\text{-}CQE}(\{ C_i \text{ set} \mid 2.6 \leq C_i.mos\text{-}cqe \leq \min(MOS_{ap}) \}))$
17	$S \leftarrow \{ C_i, C_{i-1}, \ldots, C_1 \}$
18	$Q \leftarrow \{ C_i.option, C_{i-1}.option, \ldots, C_1.option \}$
19	**else**:
20	**if** $NUM_{CE} = 1$ **and** ($Cr \in PUSER$ **or** $CE_1 \in PUSER$):
21	$C_i \leftarrow \max_{bandwidth}(C_1, \max_{MOS\text{-}CQE}(\{ C_i \text{ set} \mid C_i.bandwidth \leq \min(B_{apcr}, B_{apce1}) \}))$
22	$S \leftarrow \{ C_i, C_{i-1}, \ldots, C_1 \}$
23	$Q \leftarrow \{ C_i.option, C_{i-1}.option, \ldots, C_1.option \}$
24	**else:**
25	$DEF_{min} \leftarrow \min_{bandwidth}(DEFset_{ap})$
26	$bw_{mos} \leftarrow \max_{MOS\text{-}CQE}(DEF_{min}).bandwidth$
27	$bw_{avg} \leftarrow \text{avg}_{bandwidth}(DEF_{min})$
28	**if** $bw_{mos} < \min(B_{ap})$ **and** $(\min(REQ_{ap} - NUM_{USERi \in ap}) + 1) \times bw_{avg} \leq \min(B_{ap})$:
29	$DEF_{mos} \leftarrow \text{sort}_{MOS\text{-}CQE,Desc}(DEF_{min})$
30	S $\leftarrow DEF_{mos}$
31	$Q \leftarrow DEF_{mos}.option$
32	**else:**
33	S $\leftarrow DEF_{min}$
34	$Q \leftarrow DEF_{min}.option$
35	**end if**
36	**end if**
37	**end if**

C. mVoIP QoS Adapting Agent

The mVoIP QoS adapting agent applies the new codec preference and quality options to the IP PBX system. Fig. 3 shows the interaction between the agent and the IP PBX system in detail. In Fig. 3, Asterisk [16] is used as an IP PBX server, as it is the most popular IP PBX. The agent communicates with Asterisk Manager Interface (AMI) to maintain the action table.

The mVoIP QoS manager gathers information, such as the supported codec list, CSSR, and CDR, from the IP PBX system and updates the codec priority of each VoIP user and codec quality options to the IP PBX system through the mVoIP QoS adapting agent.

IV. Concluding Remarks and Future Work

This paper discusses previous studies of VoIP service and its performance in detail and proposes the adaptive mVoIP service architecture based on network QoS and mVoIP QoE data in SDN networks to improve mVoIP QoS. In this architecture, two key approaches are proposed: flow path optimization for mVoIP traffic using the SDN controller and the adaptive codec selection method. In particular, this paper focuses on a network condition-aware codec selection method and proposes an algorithm for adaptive codec selection for mVoIP in SDN networks.

In the future, the performance of the proposed algorithm will be evaluated in real networks and flow path optimization according to the network condition, which is the other approach, will be studied between APs and an IP PBX system in SDN networks.

Acknowledgment

This research was supported by the NIA (National Information Society Agency), the MSIP (Ministry of Science, ICT & Future Planning), Korea, under IT/SW Creative research program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-(H0502-13-1099)), the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation (2012H1B8A2025942), and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A2006331).

References

[1] B. Goode, "Voice Over Internet Protocol (VoIP)," Proc. the IEEE, vol. 90, issue 9, Sep. 2002, pp. 1495-1517.

[2] Open Networking Foundation (ONF), Software-Defined Networking: The New Norm for Networks, 13 Apr. 2012.

[3] A. Sfairopoulou, B. Bellalta, and C. Macian, "How to Tune VoIP Codec Selection in WLANs?," IEEE Communication Letter, vol. 12, issue 8, Aug. 2008, pp. 551-553.

[4] B. Bellalta, C. Macian, A. Sfairopoulou, and C. Cano, "Evaluation of Joint Admission Control and VoIP Codec Selection Policies in Generic Multirate Wireless Networks," Proc. IEEE International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 07), St. Petersburg, Russia, 10-14 Sep. 2007, pp. 342-355.

[5] S. L. Ng, S. Hoh, and D. Singh, "Effectiveness of Adaptive Codec Switching VoIP Application over Heterogeneous Networks," Proc. International Conference on Mobile Technology, Application and Systems, Guangzhou, China, 15-17 Nov. 2005, pp. 1-7.

[6] L. Roychoudhuri and E. S. Al-Shaer, "Adaptive Rate Control for Real-time Packet Audio Based on Loss Prediction," Proc. IEEE Global Telecommunications Conference (GLOBECOM 04), vol. 2, 29 Nov.-3 Dec. 2004, pp. 634-638.

[7] Z. Qiao, L. Sun, N. Heilemann, and E. Ifeachor, "A New Method for VoIP Quality of Service Control Use Combined Adaptive Sender Rate and Priority Marking," Proc. IEEE International Conference on Communications (ICC 04), vol. 3, 20-24 Jun. 2004, pp. 1473-1477.

[8] R. Sherwood, G. Gibb, K. Yap, G. Appenzeller, M. Casado, N. McKeown, and Guru Parulkar. FlowVisor: A Network

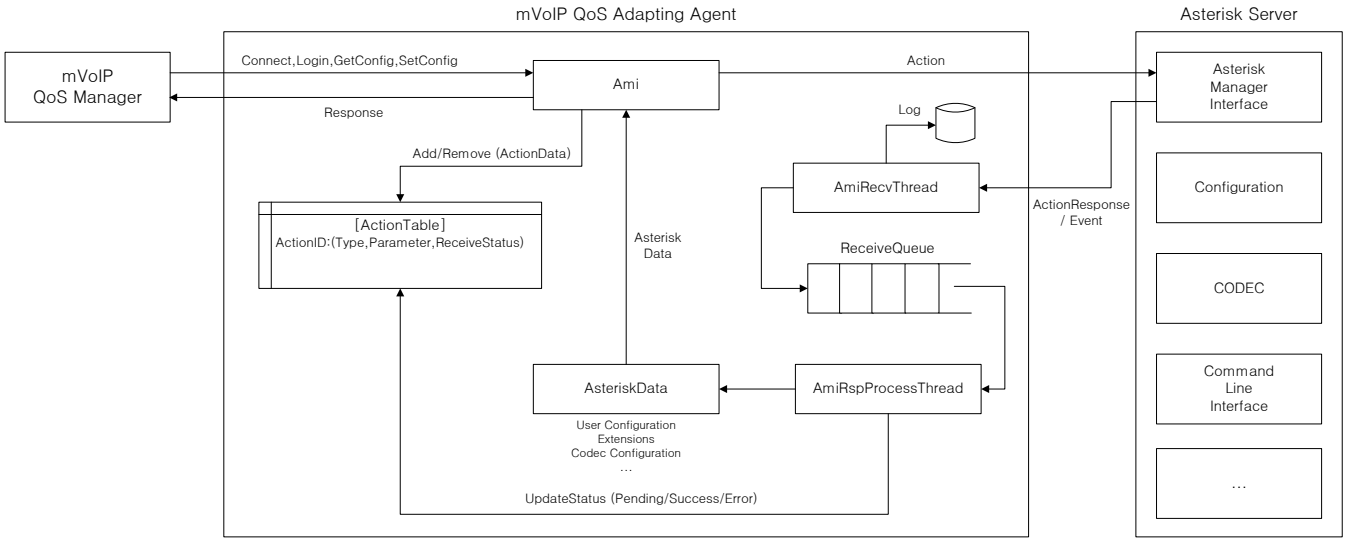


Figure 3. mVoIP QoS Adapting Agent

Virtualization Layer [Online]. Available: http://openflow.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf [retrieved: 12, 2013]

[9] A. Takahashi, H. Yoshino, and N. Kitawaki, “Perceptual QoS Assessment Technologies for VoIP,” IEEE Communication Magazine, vol. 42, issue 7, Jul. 2004, pp. 28-34.

[10] K. Kim and Y. J. Choi, “Performance Comparison of Various VoIP Codec in Wireless Environments,” Proc. ACM International Conference on Ubiquitous Information Management and Communication (ICUIMC 11), Seoul, Korea, 21-23 Feb. 2011, pp. 1-10.

[11] L. Ding and R. A. Goubran, “Speech Quality Prediction in VoIP Using the Extended E-Model,” Proc. IEEE Global Telecommunications Conference (GLOBECOM 03), vol. 7, 1-5 Dec. 2003, pp. 3974-3978.

[12] ITU-T G.107, The E-model: A Computational Model for Use in Transmission Planning, Dec. 2011.

[13] VoIPmonitor [Online]. Available: http://www.voipmonitor.org/ [retrieved: 12, 2013]

[14] pjsip-perf [Online]. Available: http://www.pjsip.org/ [retrieved: 12, 2013]

[15] Telecommunication Technology Association (Korea), QoS Measurement Methodology for Mobile VoIP, Dec. 2010.

[16] Asterisk [Online]. Available: http://www.asterisk.org/ [retrieved: 12, 2013]

 ISBN: 978-1-61208-318-6

Analytical Modelling of ANCH Clustering Algorithm for WSNs

Morteza M. Zanjireh, Hadi Larijani, Wasiu Popoola, and Ali Shahrabi
School of Engineering and Built Environment
Glasgow Caledonian University
Glasgow, UK
{Morteza.Zanjireh, H.Larijani, Wasiu.Popoola, A.Shahrabi}@gcu.ac.uk

***Abstract*—Wireless sensor networks are a popular choice in a vast number of applications, despite their energy constraints, due to their distributed nature, low cost infrastructure deployment and administration. One of the main approaches for addressing the energy consumption and network congestion issues is to organise the sensors in clusters. The number of clusters and also distribution of *Cluster Head*s are essential for energy efficiency and adaptability of clustering approaches. ANCH is a new energy-efficient clustering algorithm proposed recently for wireless sensor networks to prolong network lifetime by uniformly distributing of *Cluster Head*s across the network. In this paper, we propose an analytical method to model the energy consumption of the ANCH algorithm. The results of our extensive simulation study show a reasonable accuracy of the proposed analytical model to predict the energy consumption under different operational conditions. The proposed analytical model reveals a number of implications regarding the effects of different parameters on the energy consumption pattern of the ANCH clustering algorithm.**

***Index Terms*—Wireless Sensor Networks, Clustering, Energy Efficiency, ANCH, Analytical Model.**

I. INTRODUCTION

Wireless Sensor Network (WSN) is a network of tiny and on-board battery operated sensors with limited power of processing and radio transferring data. They can collect and send their sensed data to a base station for monitoring a remote area and perhaps to send the collected data to a remote centre. WSNs can be employed in different applications, because of their low-cost and adaptable nature, including health-care, emergency response, business, and weather forecasting [1]–[3]. Moreover, WSNs can be used in an ad-hoc manner and in harsh environments in which the attendance of human being is hard or impossible [4], [5].

Energy efficiency is essential for wireless sensor networks lifetime because there is usually no opportunity for a battery replacement or recharging. Therefore, developing energy-efficient algorithms is of higher importance in wireless sensor networks. A large amount of research has been conducted over the past few years to optimise the energy consumption in this area [6]–[8].

Clustering is a widely accepted approach for organising high number of sensors spread over a large area in an ad-hoc manner [9]. This is more useful when we consider that in most cases, neighbouring sensors sense similar data. If each sensor directly sends its data to the base station using long-distance transmission, its energy drains quickly. Moreover, this might also lead to some other issues, such as traffic congestion and data collision.

Appropriate number and size of the clusters is essential for increasing the network lifetime. For a low number of clusters, a large amount of the energy is consumed to send data from Cluster Members (CMs) to Cluster Heads (CHs). On the other hand, if the number of clusters is high, a large number of the CHs will be elected and consequently a large number of nodes will operate using long-distance transmission to communicate with the base station. Therefore, a trade-off should be made between these two factors to optimise energy consumption across the network [10].

Over the past few years, a number of clustering algorithms have been proposed. Hence, it is critical that when proposing a new algorithm, we specify its scope and evaluate it with accurate modelling of the underlying organisation and communication mechanisms. Clearly, after using such models, a comprehensive understanding of the factors that affect the potential performance of a network emerges and this makes it easier to evaluate different algorithms and select the best one for practical implementation. Employing physical experiments is impractical for a large number of configurations and running a network simulator for a large number of configurations needs an unacceptable amount of time. Analytical modelling, in contrast, offers a cost-effective and versatile tool that can help to assess the performance merits of an algorithm [11], [12].

Avoid Near Cluster Head (ANCH) is a new energy efficient clustering algorithm proposed recently for wireless sensor networks to prolong network lifetime by uniformly distributing the CHs [8]. In this paper, an analytical model for predicting the energy consumption of ANCH is proposed. The model details the affecting factors and analyses the energy consumptions under various operational conditions. The accuracy of the proposed model is evaluated using simulation.

The remainder of this paper is organised as follows. In Section II, related work is discussed. The ANCH clustering algorithm is briefly presented in Section III. The proposed analytical model of ANCH and its validation are presented in Section IV and Section V, respectively. Finally, Section VI contains our concluding remarks.

 ISBN: 978-1-61208-318-6

II. Related Work

Over the past few years, a number of clustering algorithms for WSNs have been proposed such as Low Energy Adaptive Clustering Hierarchy (LEACH) [6], Hybrid Energy-Efficient Distributed (HEED) [13], and ANCH [8]. One of the most popular clustering algorithms for wireless sensor networks is LEACH. Popularity of LEACH is not only because of its simplicity, but also for the idea of rotating CHs to efficiently balance energy consumption among nodes [6].

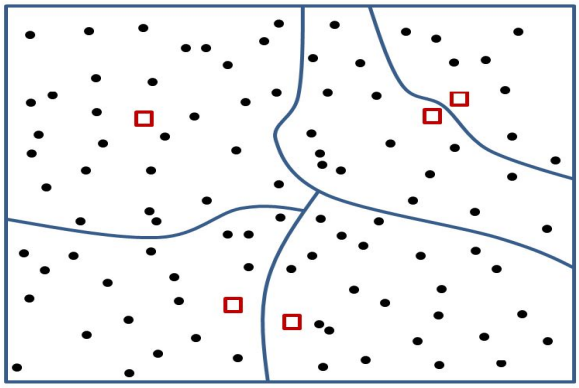


Fig. 1: An example of CHs and CMs arrangement in the LEACH algorithm.

HEED [13] is a distributed clustering algorithm for WSNs which takes into account a mixture of sensors residual energy and communication cost during CH election. In HEED, the transmission power of every node is set to a constant value and each sensor considers other nodes as its neighbouring nodes if they are within its transmission range. Moreover, two neighbouring sensors, which are within the transmission range of each other, are not elected as CH simultaneously, trying to uniformly distribute CHs across the network.

ANCH also, similar to HEED, takes the advantage of uniformly distribution of CHs in order to achieve optimised, or close to, network energy consumption. Nevertheless, it has a few key advantages over HEED. Firstly, the set-up phase overhead of ANCH is much less than that of HEED because HEED executes a procedure to find neighbouring sensors. Also, in this phase, each sensor in HEED executes a complicated iteration including some message passing to select its CH. Secondly, by the end of each iteration in HEED, a node elects itself as a CH if no other CH advertisement has been received. Thus, in many rounds, the number of formed clusters is much more than that of ANCH algorithm where all sensors receive CH advertisement if there exists at least one CH in the network. Finally, ANCH and LEACH are two scalable algorithms both with processing time and message exchange complexity of O(1) and O(N), respectively [14]. Whereas, HEED has O(N) complexity for both processing time and message exchange complexity [15], [16].

In order to design an energy efficient algorithm for wireless sensor networks, it is important to make a trade-off between different parameters involved in a specific application to ensure that the optimum configuration has been applied to maximise network lifetime. In particular, it is quite critical to balance the energy costs of individual nodes in order to obtain the best overall network energy cost. Simulation study of the effects of different parameters on the performance of a network under various network circumstances is difficult because of the time consuming feature of these kinds of tools. Analytical modelling, in contrast, is beneficial as it offers a cost-effective tool to estimate the network energy consumption accurately within an acceptable amount of time. Therefore, in addition to the research on proposing efficient algorithms for wireless sensor networks, a number of studies have also been conducted to develop analytical models [10]–[12], [17], [18].

The first analytical model for the LEACH algorithm has been proposed by Heinzelman *et al.* [17]. In this study, it has been shown that the energy consumption in a network is proportional to the square of transmission distance in clusters. This can be obtained for each sensor using the following expression:

$$E[d_{toCH}^2] = \rho \int_{\theta=0}^{2\pi} \int_{r=0}^{\frac{M}{\sqrt{\pi k}}} r^3 dr d\theta = \frac{M^2}{2k\pi} \qquad (1)$$

where $E[d_{toCH}^2]$ is expected square distance of sensors from their CH, $\rho = \frac{k}{M^2}$ and is called sensors'density, k is the number of clusters, and M is one side of network area.

However, some non-realistic assumptions have been made when developing the model; the area of all clusters are disc-shaped with radius r, all clusters are assumed to be formed equally, and also the area of the network is covered by these k non-overlapping clusters.

In [19], Bandyopadhyay and Coyle have proposed a mathematical model for hierarchical clustering algorithms for WSNs. They assumed that the sensors are very simple and all sensors transmit at a fixed power level. Their model analytically suggests the number of CHs at each level of clustering. They conducted a set of experiments to show the optimum number of CHs in different levels of hierarchy in dense networks, with up to 25,000 nodes. Nevertheless, their proposed model is not general enough due to a number of unrealistic assumptions on the fixed power level transmitting ability of nodes.

III. The ANCH Clustering Algorithm

The proper position of CHs is essential in energy efficiency of clustering algorithms. This has been neglected in the LEACH algorithm and consequently there might be some CHs which are located too close or too far from each other. In either case, some waste of energy might be occurred for data transferring from sensors to the base station.

To overcome this, the ANCH algorithm tries to uniformly distribute CHs across the network as much as possible. To do so, a parameter d is defined as the *closeness* depending on the region size and also network density. If two CHs are

 ISBN: 978-1-61208-318-6

found too close to each other in a particular round, closer than d, one of them should stand as the CH. Thus once the first CH is selected following normal LEACH procedure, the next potential CH checks its distance from the first CH before advertising itself to other sensors as a CH. If the distance is less than d, it cancels its decision to be a new CH in the current round and remains a CH candidate for the future rounds.

Further improvement in ANCH is also obtained by considering the optimum number of CHs through the network. This is because a number of potential CHs might cancel their decision of being a CH due to their close position to other CHs. Therefore, the number of clusters would be less than the optimum number suggested in the LEACH algorithm. This leads to the bigger cluster size and more energy consumption over the intra-cluster transmission.

This issue is addressed in the ANCH algorithm by increasing the threshold $T(n)$ and consequently increasing the number of potential CHs in each round. As a result, in every round more than p percent of sensors will be nominated as CHs, on average, to become closer to the optimum value, p, after dropping a number of them because of closeness issue. After setting the new threshold, close to p percent of sensors are eventually selected as the CHs in every round which are more uniformly distributed compared with LEACH. The new threshold, $T'(n)$, in ANCH is defined as follows:

$$T'(n) = T(n) + (1 - T(n)) \times a. \quad (2)$$

$T(n)$ is the threshold value of the LEACH algorithm [6] and a, the add-on coefficient, is a constant, whose value depends on network configuration and also on the *closeness* value, d. This value plays an essential role in the ANCH algorithm efficiency.

The ANCH algorithm significantly improves network energy consumption and, consequently, prolongs the network lifetime compared with the LEACH algorithm. An example of the positions of CHs and CMs in ANCH is shown in Figure 2. Comparing this arrangement with the one presented in Figure 1 reveals more uniformly distribution of CHs in the ANCH algorithm.

IV. ANCH Analytical Modelling

In this section, our proposed analytical model for the energy consumption in the ANCH clustering algorithm is presented. Using the model, a comprehensive understanding of the factors affecting the performance of a network emerges. Since a clustering approach is employed in the ANCH algorithm, the total network energy consumption can be derived when the energy consumed by one cluster is calculated.

Let us assume that N sensor nodes are randomly distributed in a M×M area and the number of clusters, on average, is k during the lifetime of the network. As a result, there are $\frac{N}{k}$ sensors, on average, per cluster with $(\frac{N}{k}) - 1$ sensors as CMs and also one node as the CH.

The energy required for a CM to send its data to a CH can be calculated using the following expression [6]:

$$E_{CM} = lE_{elec} + l\epsilon_{amp}d^2_{toCH} \quad (3)$$

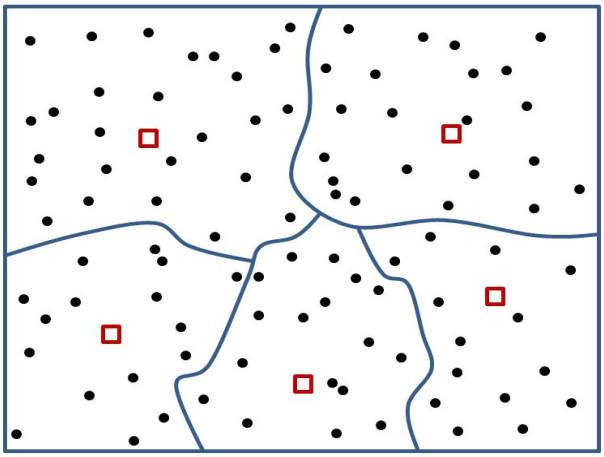


Fig. 2: An example of CHs and CMs arrangement in the ANCH algorithm.

Also, for all nodes in a cluster, this energy can be calculated as follows:

$$E_{Cluster} = lE_{elec}(k-1) + l\epsilon_{amp}E[\sum_{nodes \in Cluster} d^2_{toCH}] \quad (4)$$

where l is the length of messages, E_{elec} is the transmit electronics, ϵ_{amp} is transmit amplifier, d_{toCH} is the distance between a CM and its CH, and $E[\sum d^2_{toCH}]$ is the expected summation for square distance of CMs from their CH. Except for $E[\sum d^2_{toCH}]$, all other parameters in (4) are known with constant values. Therefore, by calculating $E[\sum d^2_{toCH}]$ we are able to calculate all the energy spent in the network.

$E[\sum d^2_{toCH}]$ can be calculated using the following expression for LEACH [20]:

$$E\left[\sum_{node \in cluster(j)} d^2_{toCH}\right] = 2\pi\lambda_{CM} \times \int_0^\infty r^3 . P\left\{(r,j) \in cluster(j)\right\} dr \quad (5)$$

In (5) and (6), λ_{CH} and λ_{CM} represent density of the CHs and CMs in the network and are given by $\frac{k}{M^2}$ and $\frac{N-k}{M^2}$, respectively. $P\left\{(r,j) \in cluster(j)\right\}$ is the probability of a sensor node to become member of cluster j. The distance between the node and the head of cluster j is also represented by r. According to [21], $P\left\{(r,j) \in cluster(j)\right\}$ can be derived from the palm distribution as follows:

$$P\left\{(r,j) \in cluster(j)\right\} = exp\left\{-\lambda_{CH}\pi r^2\right\} \quad (6)$$

In ANCH, the distance between any two CHs is not less than d. Each cluster area is divided into two different parts, which are treated separately in our model. The first part is the circular area with the radius of $d/2$ from the CH. All sensors in this area securely belong to that cluster. The second area covers those sensors whose distance from the current CH

 ISBN: 978-1-61208-318-6

is more than $d/2$. For the first part, (5) with the probability $P\{(r,j) \in cluster(j)\} = 1$ can be used. Thus, the expected summation for square distance of CMs, located in the first part of the cluster area, from their CH can be obtained using the following expression:

$$E\left[\sum_{node \in cluster(j)} d_{toCH}^2\right] = 2\pi\lambda_{CM}\int_0^{d/2} r^3 dr \quad (7)$$

On the other hand, all sensors whose distance from other CHs is less than $d/2$ are secure members of other CHs and are not members of the current CH. Thus, $P\{(r,j) \in cluster(j)\} = 0$ for those nodes. Consequently, the value of (5) for those nodes is 0. To calculate the second part of the cluster area, we must subtract the cluster areas whose nodes' distance from a CH is less than $d/2$.

The second part of each cluster area can be calculated by

$$E\left[\sum_{node \in cluster(j)} d_{toCH}^2\right] = 2\pi\lambda_{CM}\int_{R_1}^{\infty} r^3.P\{(r,j) \in cluster(j)\}\, dr \quad (8)$$

In the above expression, R_1 can be calculated as follows

$$\pi R_1^2 = k\pi(\frac{d}{2})^2 \Rightarrow R_1 = (\frac{d}{2})\sqrt{k} \quad (9)$$

Using (7) and (8), the first and second parts of each cluster area can be merged. Thus, the expected summation of square of each CM from its CH can be obtained from following expression:

$$E\left[\sum_{node \in cluster(j)} d_{toCH}^2\right] = 2\pi\lambda_{CM}. \left[\int_0^{d/2} r^3 dr + \int_{(\frac{d}{2})*\sqrt{k}}^{\infty} r^3.exp\left\{-\lambda_{CH}\pi r^2\right\} dr\right] \quad (10)$$

In Figure 3, the inner circle shows the first part of each cluster in which $P\{(r,j) \in cluster(j)\} = 1$. The area between inner and outer circles, demonstrates the first part of other clusters in which $P\{(r,j) \in cluster(j)\} = 0$. The area beyond the outer circle, shows the second part of current cluster in which $P\{(r,j) \in cluster(j)\} = exp\left\{-\lambda_{CH}\pi r^2\right\}$.

The accuracy of the proposed analytical model for ANCH is evaluated in the next section.

V. Model Validation

The accuracy of the described analytical model has been verified by comparing it with simulation results. Extensive validation experiments have been performed for several combinations of cluster size, network dimension, different values of *closeness*, density of sensors in the network, and the number of messages which are sent from CMs to their CHs during the steady phase, called *MNumbers*. In order to select parameter a, different values including $a = 0.02, 0.05, 0.15, 0.25, ..., 0.75$ have been considered and the most effective value has been selected. Each simulation scenario is run for 100 different randomly generated topologies and the average results are presented. In our experiments, the sensors' inner computational procedures do not consume energy: all of their energy used for message passing only. The energy model in all of our experiments is precisely the same as the one employed in [6].

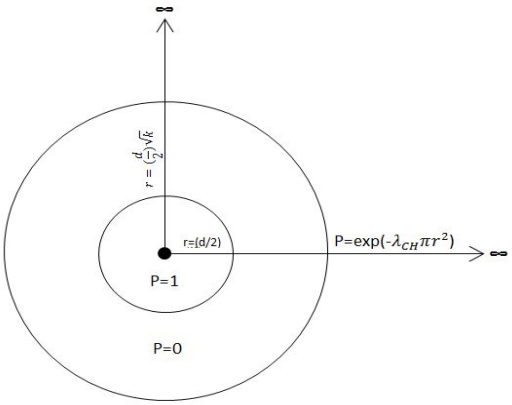


Fig. 3: An example of the first and second parts of cluster areas defined in the analytical model for ANCH.

As the first experiment, the effects of varying the number of clusters on the accuracy of our proposed model is compared against the results obtained from simulation. The network area is considered to be 50×50 square metres when base station is 100 metres away from the network's edge. Moreover, d = 15 metres and the initial energy of each node is 10 J. Finally, the number of clusters in this experiment varies from 4 to 15 clusters. The result is presented in Figure 4. In this figure, the horizontal axis shows the number of clusters where the vertical axis represents the total consumed energy. Figure 4 shows the accuracy of our model for three different networks with different number of nodes, N = 50, 100, and 200, when *MNumber* is considered to be 25. 96.3% accuracy in Figure 4 shows that the simulation results closely match those predicted by the analytical model.

In the second experiment, we aim at observing the impact of network size on our analytical model. Different network dimensions from 10 to 100 metres are examined while the value of d is 30% of one dimension. Moreover, the initial energy of each node is 10 J and the number of clusters, k, is 5. These are depicted in Figure 5, highlighting that the proposed model on average presents an accuracy of 95.4%. Figure 5 shows the accuracy of our model for three different networks with different number of nodes, N = 50, 100, and 200, when *MNumber* is considered to be 25.

In the third experiment, we aim at observing the impact of *closeness* parameter, d, on our analytical model. Different *closeness* values from 5 to 25 metres are examined where

 ISBN: 978-1-61208-318-6

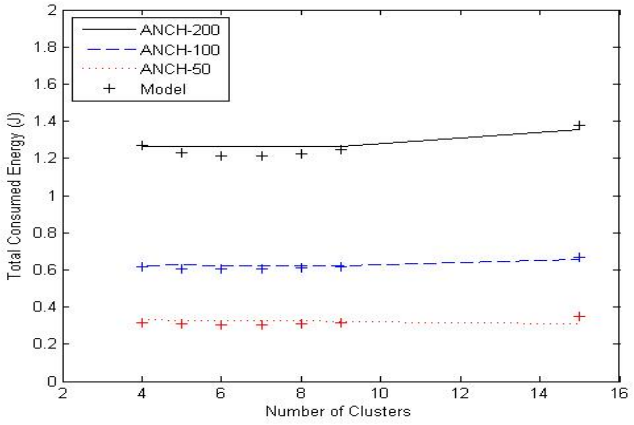


Fig. 4: Accuracy of the model comparing against simulation results varying number of clusters for three networks with different number of nodes, N=50, 100, and 200.

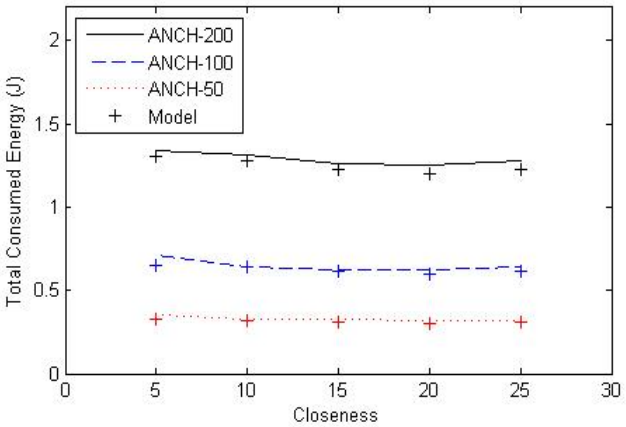


Fig. 6: Accuracy of the model comparing against simulation results varying parameter d for three networks with different number of nodes, N=50, 100, and 200.

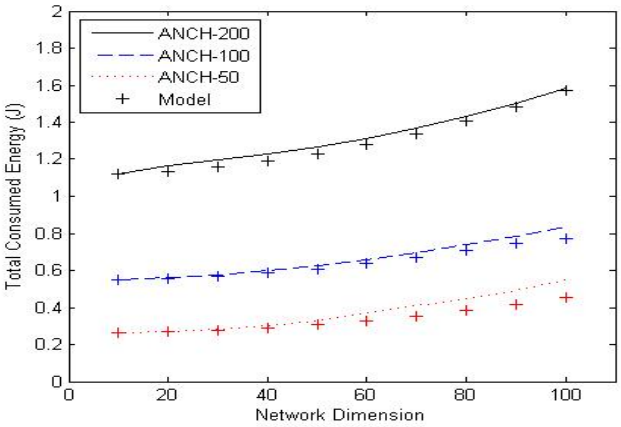


Fig. 5: Accuracy of the model comparing against simulation results varying network dimension for three networks with different number of nodes, N=50, 100, and 200.

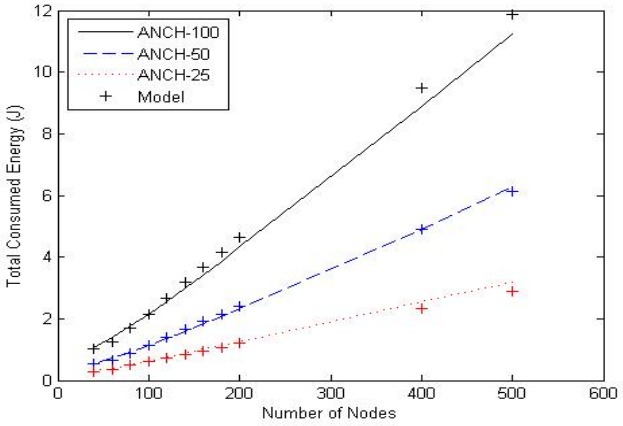


Fig. 7: Accuracy of the model comparing against simulation results for three values of *MNumber*, *MNumber* = 25, 50, and 100 messages per round.

the network area is considered to be 50×50 square metres and base station is 100 metres away from the network's edge. Moreover, the initial energy of each node is 10 J and the number of clusters is 5. This is depicted in Figure 6, highlighting very close agreement between the model and simulation in this figure, 95.8 similarities on average. Figure 6 demonstrates the accuracy of the proposed model for three different networks with different number of nodes, N = 50, 100, and 200, when *MNumber* is considered to be 25.

In the fourth experiment, we aim at observing the impact of network density on our analytical model. In this experiment, different number of sensors, from 40 to 500, are examined. Moreover, the network area is 50×50 square metres when base station is 100 metres away from the network's edge, d=15 metres, the initial energy of each node is 10 J, and the number of clusters is 5. The results are presented in Figure 7 for three different configurations, *MNumber* = 25, 50, and 100.

These results show a close agreement, an accuracy of 95.4% on average, between the proposed model and simulation results.

Finally, in the last experiment, we aim at observing the impact of steady phase duration on our analytical model by varying the number of *MNumber* from 5 to 1000 messages per round. The network area is 50×50 square metres when base station is 100 metres away from the network's edge, d=15 metres, the initial energy of each node is 10 J, and the number of clusters is 5. In Figure 8, the comparison of the model and simulation results for three different networks with N = 50, 100, and 200 nodes are presented, approving 95.6% accuracy on average.

Overall, our extensive validation study show the credible accuracy of our proposed analytical model to predict the total energy spent by the ANCH algorithm.

Using the proposed model, a number of implications have

 ISBN: 978-1-61208-318-6

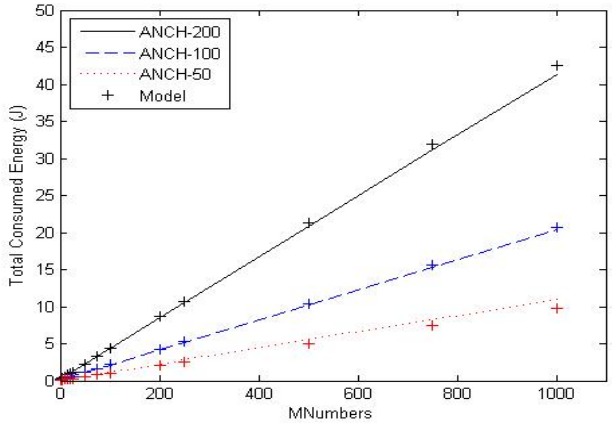


Fig. 8: Accuracy of the model comparing against simulation results for three networks with different number of nodes, N=50, 100, and 200.

been revealed. First, the energy consumed by the ANCH algorithm is almost insensitive to the optimum number of clusters, k, proposed by the LEACH algorithm. This is due to the important role of add-on coefficient, a, to balance the energy consumption of each cluster. By increasing the value of k, the optimum value of a is also increased to protect the network from forming a large number of clusters with smaller number of nodes in each cluster and hence to avoid wasting energy. Respectively, the optimum value of a is also decreased to block the negative effects of smaller number of clusters.

In the same way, the energy consumed by the ANCH algorithm is almost insensitive to *closeness* parameter. This is again due to the balancing role of add-on coefficient, a. By increasing the value of *closeness* parameter, the optimum value of a is also increased to increase the number of potential CHs to avoid smaller number of clusters. It also prevents forming large number of clusters when the *closeness* value is decreased.

VI. Conclusion

ANCH is a distributed energy-efficient clustering algorithm proposed for wireless sensor networks. ANCH prolongs the network lifetime by uniformly distributing of CHs across the network. In this paper, we have presented an analytical model for ANCH to show the effects of different parameters and to predict overall energy consumption under various network conditions. Our extensive validation study has demonstrated a reasonable degree of accuracy achieved by our analytical model compared with the results of a simulation software. The proposed analytical model has also revealed that energy consumption of the ANCH algorithm is almost insensitive to the number of clusters and *closeness* parameter due to the balancing role of add-on coefficient to optimise the total energy consumption of clusters.

References

[1] I. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, 2002.

[2] J. M. Chung, Y. Nam, K. Park, and H. J. Cho. Exploration time reduction and sustainability enhancement of cooperative clustered multiple robot sensor networks. *Network, IEEE*, 26(3):41–48, 2012.

[3] M. M. Zanjireh, A. Kargarnejad, and M. Tayebi. Virtual enterprise security: importance, challenges, and solutions. *WSEAS Transactions on Information Science and Applications*, 4(4):879–884, 2007.

[4] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie. Protocols for self-organization of a wireless sensor network. *Personal Communications, IEEE*, 7(5):16–27, 2000.

[5] C. Tselikis, S. Mitropoulos, N. Komninos, and C. Douligeris. Degree-based clustering algorithms for wireless ad hoc networks under attack. *Communications Letters, IEEE*, 16(5):619–621, 2012.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 1–10, 2000.

[7] A. Maizate and N. El Kamoun. Efficient survivable self-organization for prolonged lifetime in wireless sensor networks. *International Journal of Computer Applications*, 58(16):31–36, 2012.

[8] M. M. Zanjireh, A. Shahrabi, and H. Larijani. Anch: A new clustering algorithm for wireless sensor networks. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 450–455, 2013.

[9] I. Baidari, H. B. Walikar, and S. Shinde. Clustering wireless sensor and wireless ad hoc networks using dominating tree concepts. *International Journal of Computer Applications*, 58(20):6–9, November 2012.

[10] A. Al Islam, C. S. Hyder, H. Kabir, and M. Naznin. Finding the optimal percentage of cluster heads from a new and complete mathematical model on leach. *Wireless Sensor Network*, 2(2):129–140, 2010.

[11] L. S. Bai, R. P. Dick, P. H. Chou, and P. A. Dinda. Automated construction of fast and accurate system-level models for wireless sensor networks. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1–6, 2011.

[12] I. Beretta, F. Rincon, N. Khaled, P. R. Grassi, V. Rana, and D. Atienza. Design exploration of energy-performance trade-offs for wireless sensor networks. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, pages 1043–1048, 2012.

[13] O. Younis and S. Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3(4):366–379, 2004.

[14] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich. A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 15(1):121–144, 2013.

[15] C. Li, M. Ye, G. Chen, and J. Wu. An energy-efficient unequal clustering mechanism for wireless sensor networks. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, 2005.

[16] Q. Zhang, R. H. Jacobsen, and T. S. Toftegaard. Bio-inspired low-complexity clustering in large-scale dense wireless sensor networks. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 658–663, 2012.

[17] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, 2002.

[18] J. Gupchup, A. Terzis, R. Burns, and A. Szalay. Model-based event detection in wireless sensor networks. In *Workshop on Data Sharing and Interoperability on the World-Wide Sensor Web (DSI)*, 2008.

[19] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1713–1723, 2003.

[20] J. C. Choi and C. W. Lee. Energy modeling for the cluster-based sensor networks. In *Computer and Information Technology, 2006. CIT '06. The Sixth IEEE International Conference on*, pages 218–218, 2006.

[21] S. Foss and S. Zuyev. On a certain segment process with voronoi clustering. 1993.

 ISBN: 978-1-61208-318-6

Survivability Mechanism for Multicast Streaming in P2P Networks

Rober Mayer, Manoel C. Penna, Marcelo E. Pellenz and Edgard Jamhour
Graduate School of Computer Science
Pontifícia Universidade Católica do Paraná - PUCPR
Rua Imaculada Conceição, 1155, 80215-901, Curitiba, Brazil
email: {rober, penna, marcelo, jamhour}@ppgia.pucpr.br

***Abstract*—This paper proposes a mechanism for the survivability of multicast streams on Peer-to-Peer (P2P) overlay networks, capable to protect the network against failures of source and intermediary nodes. Based on a mixed integer programming model, an algorithm is proposed to build the main multicast tree and the necessary backup trees. The mechanism also includes a recovery protocol that identifies the failures and recovers the multicast stream by activating the appropriate backup tree. The proposed mechanism is evaluated by means of numerical experiments, which demonstrate its effectiveness with respect to the quality of the backup multicast trees, to recovery time and to the bandwidth overhead. It is concluded that the proposed mechanism can improve the reliability of multicast streams on static P2P networks with efficiency and quality.**

***Keywords*—*Static P2P Networks*, *Resilient P2P Multicast*.**

I. INTRODUCTION

Multicast transmission is an efficient approach for supporting applications involving group communication. The main characteristic of multicast transmission is its ability to eliminate most of the redundant packets necessary to the transmission to multiple destinations. It is particularly useful for video streaming applications, that is, the distribution of video streams from one source to multiple destinations [1].

Multicast protocols are responsible to route the application packets in multicast transmissions. In response to the difficulties of implementing multicast transmission in the IP network layer, several recently proposed protocols adopt the Peer-to-Peer (P2P) communication model, in which all multicast features are implemented exclusively in user hosts instead of routers [2][3]. P2P is a self adaptive communication model implemented in the application layer, where the participating peers are configured in an overlay network. Multicast protocols deployed on P2P networks, called P2P multicasting, are applications protocols that implement multicast features over the P2P unicast links.

In dynamic P2P networks, a peer joins the system when a user starts the application, contributes with some resources while making use of some resources provided by others, and leaves the system when the user exits the application. Such join-participate-leave cycle is a key characteristic in dynamic P2P networks. The independent arrival and departure of peers create a collective effect called churn. On the other hand, in static P2P networks, all participating peers are interested in staying in the system. Consequently, they do not leave the application as frequently as in dynamic P2P networks, reducing considerably the churn effect. In this case, a peer leaving the system can be considered as a failure event. Examples of static P2P multicasting applications are streaming systems applied for dissemination of critical information, Internet Protocol TeleVision (IPTV) system using set-top boxes, and content Delivery Network (CDN).

Since the transmitted content is assumed to be of interest to the applications, the P2P multicasting should provide with delivery guarantees. In general, two basic approaches can be considered to provide network survivability: protection and restoration. The first assumes that backup trees are pre-computed while the second applies dynamic routing, i.e., new backup routes are computed only when a failure occurs. The main advantage of protection is quick reaction to failures, usually providing smaller recuperation time.

The aim of this work is to propose a protection based survivability mechanism for multicast streams on static P2P networks, capable to protect P2P multicasting against failures of both the source peer and intermediary peers. In the first case, a traditional model of flow conservation is used to construct a set of multicast trees, one to be used as the main tree, and the others capable of protecting the multicast stream in case of failure of the source peer. The model is extended to build another set of backup trees that will be used to recover from failures of intermediary peers. A recovery protocol is also necessary in this case, to detect eventual failures and recover the streaming application by switching the transmission to the appropriate backup tree.

The main objective of this study is to propose and validate the recovery protocol. For this reason, in the current state of our proposal, the backup trees are computed off-line according to a model of Mixed Integer Programming (MIP). Although the MIP model provides the optimal set of backup trees, it severely limits the size of the network. We are developing a heuristic algorithm to generate dynamically the backup trees, but it discussion is not in the scope of this work.

A. Motivation and Contributions

In this study, it is assumed that the content of the flows requires delivery guarantees, and thus, survivability mechanisms are necessary. It is also assumed that the participating peers do not leave the streaming application voluntarily and independently. The study is motivated by the growing needs for fast deployment of this kind of streaming services in a reliable and cost effective way. Examples of applications requiring resilient multicast streams in static P2P scenarios

 ISBN: 978-1-61208-318-6

are weather forecasts, security notifications, software updates, and traffic information.

Most of previous research concentrates on recovering dynamic P2P networks [4]. They focus on finding many disjoint parents for each peer and on the protocols to recover the multicast routes after the occurrence of events caused by a peer failure or by the churn effect. Usually in this case, the path diversity is reduced by assuming that the set of ancestors of a node in each tree should be as disjoint as possible.

Some few studies consider the resilience of the multicast tree in static P2P networks by applying resilience mechanisms of kind 1:1, in which streaming flows are delivered simultaneously through multiple multicast trees rooted in different source peers [7][10]. Because the recovery actions switch the source peer and the multicast tree when a failure occurs, recovery protocols are not necessary.

In this article, P2P multicasting is investigated at the level of streams as well at the level of network control. At the stream level we present extensions to an integer programming model used to compute the multicast trees. At the network control level we specify and evaluate a recovery protocol. On the other hand, we do not deal neither with multicast streams at the packet level, nor with the join and leave procedures of peer nodes, because the assumption of static P2P network. Note, however, that the proposed method can be applied in conjunction with other proactive and/or reactive methods proposed literature. To the best of the author's knowledge, there is no study addressing the recovery of intermediary node failures in static P2P multicast trees.

B. Organization of the Paper

The remaining of the paper is organized as follows: In Section 2, we present previous research on P2P multicasting with a special focus on survivability. Section 3 presents the proposed recovery mechanism, including the protection model (the mixed integer programming model) and the recovery protocol. In Section 4, the proposed scheme is evaluated, and the results compared with other published studies. Finally, the last section concludes the paper.

II. Related Work

A review of resilient approaches to peer-to-peer multicast is presented by Hongyun et al. [4], where they are organized into three classes, according to overlay construction approach: tree-based, mesh-based and data-driven. Tree based approaches organize the participating peers into a single or multiple logical trees over which the multicast data is transmitted. The studies in [5], [6] and [7] are examples of studies that can be included in this class. In mesh-based P2P multicast the participating peers form a mesh overlay network and the stream content is delivered through routing protocols. An example included in this class is [8]. In data driven protocols, the necessity of data defines the streaming routing, that is, a peer always forwards data to others that are expecting for it. An example in this class is [9].

Our approach can be classified as a tree-based approach, and will be compared to [6], where Probabilistic Resilient Multicast (PRM) is introduced. It is a multicast recovery scheme based in two basic components. The first is a proactive component in which each peer randomly selects a constant number of other peers, forwarding data to each of them with a small probability. This random forwarding occurs in parallel to the usual data forwarding along the multicast tree, leading to a small number of duplicate packets, which are properly detected and suppressed. The second is a reactive mechanism to handle eventual data losses. According to the authors, these mechanisms can together provide high resilience guarantees and can be used to significantly improve the data delivery ratios of application-layer multicast protocols.

In our proposal, multicast trees and backup trees are computed off-line according to a model of Mixed Integer Programming (MIP). We define extensions to the work proposed by Walkowiak et al. [10] [7]. In the first, the authors addressed multicast transmission in static P2P networks by formulating an optimization problem that builds disjoint multicast trees to protect the system against failures of root nodes and Internet Service Provider (ISP) links. The goal is to minimize streaming costs and maximize transmission throughput. Walkowiak et al. [7] presented an extension of the previous work by also considering the delay metric. By evaluating the results from the proposed MIP model they concluded that the P2P multicasting systems can be enhanced with additional protection methods, without significant reduction of system performance. They also presented a heuristic algorithm to solve the problem.

Our proposal differs from [6] because the reconfiguration of routes in the multicast trees is not probabilistic, but based on optimal backup trees computed off-line. Unlike the proposals found in [7] and [10], our approach provides resilience from backup trees that are activated in the event of failures, and not through the simultaneous transmission of multiple streams through the trees for protection.

III. Recovery Mechanism

In this paper, we consider node failures and loss of control messages. Three types of failures are considered and explained in the following: the root node, leaf node and intermediate node. The loss of control messages are handled by the recovery protocol through KEEPALIVE messages and timeout mechanism. Flows are propagated through the multicast tree, and the rooting peer is responsible for the transmission. Intermediary peers are responsible for forwarding the flows and leaf peers only consume the flows. Intermediary peers are also flow consumers. Failure of the root peer is critical, since it completely interrupts flow transmission. To solve this problem one can use multiple backup trees with different roots. In case of failure, the resilience is ensured by switching the responsibility transmission to the root of a backup tree. The failure of intermediary peers is also critical, because all its successors are disconnected from the tree. The solution here is to use multiple backup trees having the same root. The backup

 ISBN: 978-1-61208-318-6

```
1: for each (ROOT in the problem) do
2:     Compute the main tree MT rooted in ROOT according to the standard MIP model
3:     for each (failed peer FP in MT) do
4:         Compute the backup tree BT rooted in ROOT according to the modified MIP model
5:     end for
6: end for
```

Figure 1. Main tree and backup tree computing

trees exclude the failed intermediary node, and are activated only when the corresponding failure occurs. Handling the failure of a leaf node is trivial. Nothing needs to be done because it only leaves a vacant place in the lowest hierarchy.

Failures of root peers are addressed by providing multiple main trees routed on different nodes. Main trees are calculated by using the MIP model proposed by Walkowiak et al. [7], referred in this paper as the standard MIP model. Failures of intermediary nodes need an extension to the standard MIP model in order to allow the construction of backup trees that exclude the failed peer. Also, a recovery protocol is needed. The advantage of this approach is to avoid the waste of bandwidth caused by simultaneous transmissions across multiple main trees, but requires the recovery protocol.

The computation of backup trees is performed before the streaming application starts. The lowest cost backup trees are calculated and stored in recovery tables. Each peer stores a recovery table for each possible intermediary node failure. Because the calculation of recovery tables is performed off-line, the computing time is not critical. The recovery mechanism, named tree switching recovery (TSR) is described is the sequence.

A. Computing the Main Tree and the Backup Trees

A set of multicast trees (main trees) are computed, for the original root and for each backup root, according to the standard MIP model. To protect the streaming application against failures of intermediary nodes, for each main tree, a set of backup trees are computed. For this, the standard MIP model is modified to prevent the failed node to be present in the solution. The modified MIP model is then applied considering the exclusion of every intermediary node present in the main tree, as presented in the algorithm of Figure 1. When more than one failure occurs, the modified MIP model should be executed for each failure combination. The standard MIP model can be reduced to the hop-constrained minimum spanning tree problem, which is NP-complete. This severely limits the size of the network and a heuristic method for backup tree generation is necessary, but it is out of the scope of this study.

B. Backup Trees and Recovery Tables

Multicast trees are identified according to its root peer and failed peer. Assuming that there are R root peers, and I intermediary nodes, main trees and backup trees are identified by $T_{i,j}$, where i is the identifier of the root peer $(i = 1, ..., R)$, and j is the identifier of the failed peer $(j = 0, ..., I)$. When no intermediary node is failed, $j = 0$. For example, $T_{1,0}$ identifies the main tree rooted in 1 and $T_{5,3}$ identifies the backup tree rooted in 5 with the intermediary peer 3 failed.

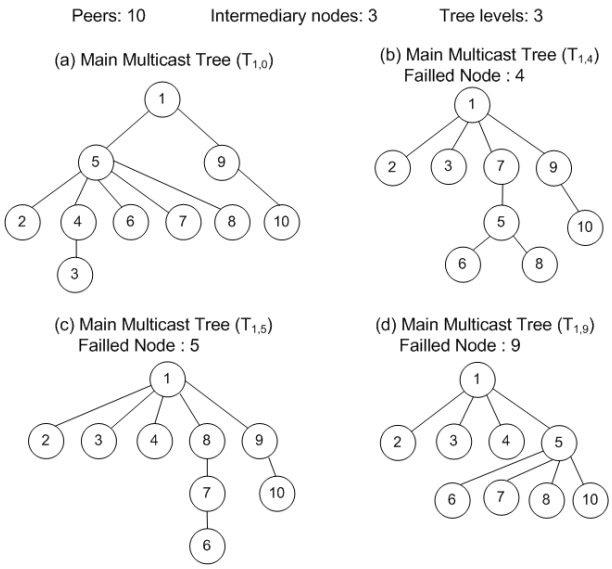


Figure 2. Example of Main Tree and Backup Trees.

Figure 2 illustrates the multicast trees for a network with ten peers, limited to three levels. The root peer is 1 and there are three intermediary nodes (4, 5 and 9). The main multicast tree $T_{1,0}$ is depicted in Figure 2a, and backup trees $T_{1,4}$, $T_{1,5}$ and $T_{1,9}$ are depicted in Figures 2b, 2c and 2d, respectively.

Recovery tables are identified by R_i, where i is the identifier of the root node. The recovery table of the tree rooted at peer 1 (R_1) in the example of Figure 2 is shown in Table I. Recall that it will be necessary one table for each root peer. As it can be seen, the recovery tables are represented by a matrix structure, where the columns represent the failed peer (0 representing no failures) and the rows represent a peer. Each cell stores at the list of children of the peer in backup tree.

TABLE I . RECOVERY TABLE EXAMPLE

Failure / Node	0	4	5	9
1	5,9	2,3,7,9	2,3,4,8,9	2,3,4,5
2	-	-	-	-
3	-	-	-	-
4	3	-	-	-
5	2,4,6,7,8	6,8	-	6,7,8,10
6	-	-	-	-
7	-	5	6	-
8	-	-	7	-
9	10	10	10	-
10	-	-	-	-

C. Recovery Protocol

The recovery protocol includes fault detection and recovery mechanism. Fault detection is based on exchanging KEEPALIVE messages and on a timeout scheme with retries. Every peer (except the root) periodically sends a KEEPALIVE

 ISBN: 978-1-61208-318-6

message to its parent. After a period of time without receiving KEEPALIVE messages from one of its children (timeout expiration) and having exhausted the number of retries, a fault is identified. The peer then switches to the backup tree corresponding to the failed peer and sends a reconfiguration message to his father. The message is sent recursively until it reaches the root, which in turn sends an activation message down to their children, informing the recovery table to be activated, which is recursively propagated to the leaf peers.

The recovery protocol is a multi-threaded algorithm. Thread 1, described in the algorithm of Figure 3, implements the recovery procedure, while Threads 2 and 3 (showed in the algorithms of Figures 4 and 5) are responsible for fault detection. The following global variables are defined: the identification of the peer executing the protocol (PiD), the current root of multicast tree (Root), the current recovery table ($RecoveryTable$). Each child peer has a timer (Timer) and a counter of the number of retries (NoR). All messages have three fields, the sender id, the message type and the identification of the failed peed (FiD), if it is the case. Three types of messages are defined: KEEPALIVE, RECOVER and ACTIVATE.

Thread 1 synchronously receives messages (Line 2). For KEEPALIVE messages it just restarts the timer end the number of retries of the corresponding child. When a RECOVER message is received (Line 7), the peer activates the recovery table of the faulty child (Line 8), then propagates a RECOVER message to it is parent (except the root peer) and waits for an ACTIVATE message (Lines 15-17). When the ACTIVATE message arrives, the peer activates the recovery table of the faulty peer (Line 26) and sends an ACTIVATE message to each of its children (Lines 30-34).The flow of RECOVER and ACTIVATE messages corresponds to a tree traversal, with the computational complexity equals to $O(\log n)$.

Thread 2 sends KEEPALIVE messages. Line 10 defines a PAUSE interval between two KEEPALIVE messages. This is a parameter of the recovery protocol, and its setting is discussed in section IV.

Thread 3 is invoked when a timeout occurs for one of the children (in Child variable) of the peer executing the protocol. Line 2 defines a LIMIT on the number of retries before considering Child has failed. This LIMIT is also a parameter discussed in section IV. In line 8 the peer sends a RECOVER message to itself, informing the failure of Child.

```
1: while TRUE do
2:     RECEIVE (Message)
3:     if Message.Type = KEEPALIVE then
4:         Child = Message.Sender
5:         Timer.Child = time_now + TIMEOUT
6:         NoR.Child = 0
7:     else if Message.Type = RECOVER then
8:         Activate the RecoveryTable for Messge.FiD
9:         if PiD ≠ Root then
10:            Retrive Parent in the active RecoveryTable
11:            Message1.Sender = NiD
12:            Message1.Type = RECOVER
13:            SEND (Message1) to Parent
14:            RECEIVE (Message1)
15:            while Message1.Type ≠ ACTIVATE do
16:                RECEIVE (Message1)
17:            end while
18:        end if
19:        Retrieve Children in active RecoveryTable
20:        Message1.Sender = NiD
21:        Message1.Type = ACTIVATE
22:        for each Child in Children do
23:            SEND (Message1) to Child
24:        end for
25:    else if Message.Type = ACTIVATE then
26:        Activate the RecoveryTable for Message.FiD
27:        Retrieve Children in active RecoveryTable
28:        Message2.Sender = PiD
29:        Message2.Type = ACTIVATE
30:        for each Child in Children do
31:            SEND (Message2) to Child
32:            Timer.Child = time_now + TIMEOUT
33:            NoR.Child = 0
34:        end for
35:    end if
36: end while
```

Figure 3. Recovery Protocol - Thread 1

```
1: if (PiD = Root) then
2:     exit
3: end if
4: while TRUE do
5:     Retrieve Parent in the active RecoveryTable
6:     Message.Sender = PiD
7:     Message.Type = KEEPALIVE
8:     Message. FiD = NONE
9:     SEND (Message) to Parent
10:    SLEEP(PAUSE)
11: end while
```

Figure 4. Recovery Protocol - Thread 2

```
1: NoR.Child = NoR.Child + 1
2: if (NoR.Child < LIMIT) then
3:     exit
4: else
5:     Message.Sender = PiD
6:     Message.Type = RECOVER
7:     Message.FiD = Child
8:     SEND (Message) to PiD
9: end if
```

Figure 5. Recovery Protocol - Thread 3

IV. Evaluation

In this section we evaluate the performance of the proposed resilience mechanism. The evaluation of the application requirements is outside the scope of this study. The resilience mechanism is evaluated according to three distinct aspects: quality of backup trees, recovery time, and bandwidth overhead. The quality of the backup trees is evaluated with respect to their costs. In the standard MIP model, the cost of a solution is given by the delays introduced in all branches of a tree. The same metric is used to evaluate the quality of backup trees, whose costs are compared to the main tree.

To evaluate the recovery time and the bandwidth overhead, the recovery protocol was implemented in an event driven simulator, based on the actors-messages paradigm. According to this paradigm, a simulation model is composed by a set of actors or tasks that communicate among them using messages. Nodes are implemented as task and communication channels are implemented as queues. Regarding the topology of the evaluated networks, it was assumed that all nodes are interconnected by virtual (Internet) links. An important

 ISBN: 978-1-61208-318-6

issue was to model the transmission delay in links of the overlay network. Several studies have addressed this point and the following two models for Internet link delay were considered. Hongli [11] used a maximum likelihood estimation procedure to find the best fit distribution to a large set of data measured over the Internet. They found the link delay follows the Gamma distribution. Kaune et al. [12] built a model for Internet delay based on geographical location. This model has two components, the minimum RTT and the jitter. Both models were implemented in the simulation, the first for distances less than 200 km and the other for lager distances. In this simulation scenario the actual position of the nodes is irrelevant and just the distance between each pair of nodes in the backup tree should be considered, which was modeled as a random variable uniformly distributed between 10 and 600 km. The recovery time is evaluated with respect to the amount of peers in the P2P network.

Bandwidth overhead is computed by measuring the extra bytes necessary to provide the survivability feature. We analyzed the bandwidth overhead by measuring the bandwidth consumed for the purpose of reconfiguration, that is, the average number of bytes sent per second in the multicast tree, in addition to normal transmission rate of data. For this case, TSR and PRM are compared. Considering that this study only addresses control of PDP networks, evaluations are made considering only the control plane messages. The traffic model for control messages in the evaluation scenarios is provided in the subsections that follow. All simulations are executed 30 times and the results provided for a confidence level of 99%.

A. Quality of the Backup Trees

The quality of the backup trees is assessed by the average delay costs introduced by the tree (considering all branches). Figure 6 shows the average tree costs with respect to the amount of peers. It can be observed that the costs remain close to those of the main tree, even with the modified restrictions and with one less peer in the backup tree (the failed peer). This is important, because depending on the faulty intermediary peer, the cost of the backup tree could increase due to the lower amount of available network resources.

B. Recovery Time

The recovery time evaluates how fast the recovery mechanism recovers from failures. Recovery time is composed by fault detection time and propagation time. Detection time is the time it takes for a peer to realize that one of its children has failed. Propagation time is the time it takes to the RECOVER message to arrive at the root, plus the time it takes to the ACTIVATE messages to be disseminated along the backup tree.

Detection time depends on the values of parameters PAUSE and LIMIT in the algorithms of Figures 3, 4 and 5. LIMIT is set to 3, allowing at most 2 retries. PAUSE is set as a multiple of the maximum delay time, named here the maximum delay time factor (*MDTF*). The maximum delay time is computed to cover the 95 percentile of the distribution of the delay time.

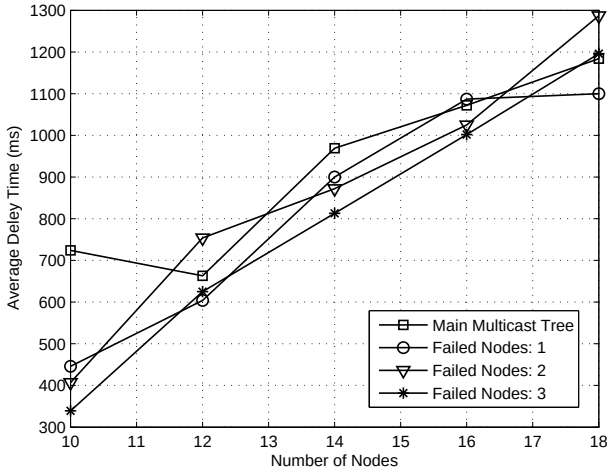


Figure 6. Quality of Backup Trees.

Figure 7 shows the normalized recovery time and the normalized bandwidth overhead for delays varying from 10 to 130ms. The normalized values are plotted for *MDTF* varying from 0.5 to 2.0. We can observe that $MDTF = 1.5$ leads to good values for both recovery time and bandwidth overhead. As a result we defined $MDTF = 1.5$, that is, the PAUSE parameter equals 1.5 times the maximum message time.

In Figure 8, we can see the average recovery time for the size of network from 10 to 18 peers. It can be observed that the recovery time remains stable as the number of peers increases. Although one could expect increasing the recovery time with the network size, since the total cost of the system increases, it remains almost constant from the fact that when the quantity of peers increases, the transmission means (upload and download links) also become more abundant. As a result, peers have more children in optimal trees, decreasing the number of levels and therefore reducing the delay. The graph in Figure 8 shows the average recovery time taken for a sample of 30 simulations, shown with 99% confidence level.

C. Bandwidth Overhead

We call bandwidth overhead, the control information transmitted by the recovery protocol, and we analyzed it by measuring the bandwidth consumed by the recovery protocol. We measured the average number of bits per second sent in the multicast tree for the transmission of control messages. In TSR, control messages are initiated by parents of intermediary peers that failed, and are targeted recursively at their new parents until reaching the root node. In PRM, recovery is achieved by the redundant probabilistic forwarding of data.

To simulate PRM behavior we used the implementation made available by Birrer [13]. We considered from 10 to 18 nodes distributed in sites of PlanetLab [14], with the parameters presented by Birrer et al. [6]. The probability of redundant transmission is set to 0.01. The amount of redundant messages exchanged between peers increases with the number of peers because PRM needs to discover and maintain the list

 ISBN: 978-1-61208-318-6

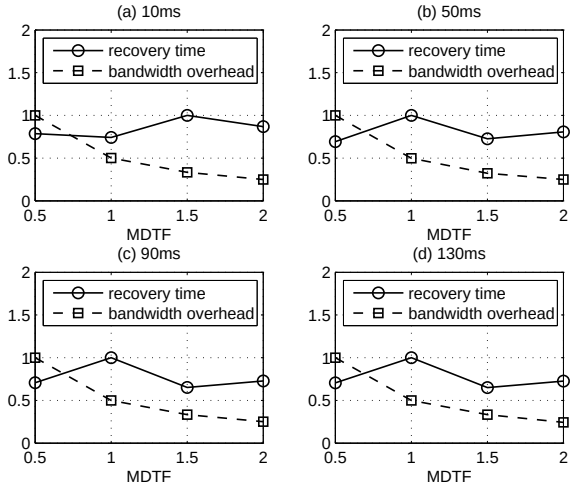


Figure 7. Normalized Recovery Time and Bandwidth Overhead.

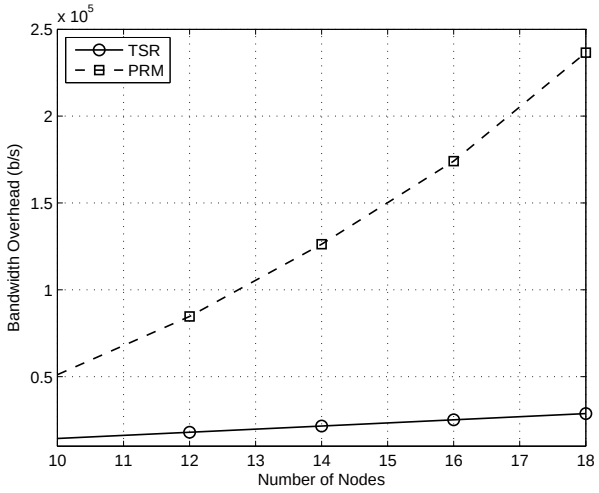


Figure 9. Bandwidth Overhead.

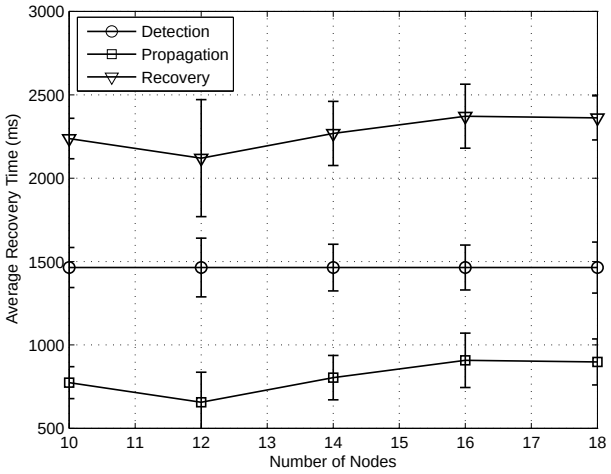


Figure 8. Recovery Time.

of pairs included in random routing. In TSR, the overhead due to control messages is negligible, because a single RECOVER message is sent by the node that detected the fault that runs through part of a branch of the tree to the root. To activate the new configuration, only one ACTIVATE message is sent to each node. On the other hand, the overhead introduced by fault detection is significant and is directly affected by the 2 parameters of the algorithm of Figure 3.

Figure 9 shows how the bandwidth overhead is affected by the size of the P2P network. In both systems (PRM and TSR) we can observe that the bandwidth overhead increases linearly with the number of peers. In PRM this is because the random forwarding of redundant packets, while in TSR the overhead is mainly due to the KEEPLIVE messages. It can be seen that as the size of the network grows, TSR achieves better results when compared to PRM. It may be noted that for 18 peers, the curve of the proposed mechanism is about 8 times lower than the PRM. The reason for this is that the overhead in TSR occurs only by the amount of KEEPALIVE messages needed to test the connectivity between network peers, and because the difference between rates in the control plane and the data plane is very large. The low increase in overhead for TSR indicates that the mechanism has good scalability with respect to bandwidth overhead.

V. Conclusions and Future Work

This paper proposed a recovery mechanism in the P2P multicasting networks named TSR. Its goal is ensure the survival of multicast flows in the presence of failures of intermediary nodes. TSR was evaluated according to three aspects: the quality of multicast tree, the recovery time, and the bandwidth overhead. The quality of backup trees is suitable due to the characteristics of the extended MIP model. It was demonstrated that, even with the additional constraints of the extended MIP model, the cost remains similar to the main multicast tree obtained with the standard MIP model. The recovery protocol was evaluated and it can be concluded that it is not severely impacted by the addition of control messages, because when a node resets the multicast tree, only a few reconfiguration messages go through a few links. The cost converges to a stable value as the number of levels in the multicast tree increases. This is because the size of recovery messages is small, because more transmission resources are made available when the number of peers increases, and because the number of levels in the multicast tree decreases when the amount of peers increase. TSR was also evaluated with respect to the bandwidth overhead, being compared to a classical reactive P2P survivable protocol. The increase of bandwidth overhead in both cases is linear, but the slope is smother in TSR. From the presented results we can conclude that the proposed mechanism improves the reliability of multicast streams on static P2P networks with efficiency and quality. However, the limitations of scale related to the size of the P2P network introduced by the MIP model must be addressed in future work.

 ISBN: 978-1-61208-318-6

REFERENCES

[1] S. Birrer and F. E. Bustamante, "Resilient Peer-to-Peer Multicast without the Cost," Twelfth Annual Multimedia Computing and Networking (MMCN), 2005, pp. 113-120.

[2] O. C. Kwon, H. Song, and T. Um, "A robust P2P video multicast streaming system under high peer-churn rate," Proceedings of the 13th International Conference on Communication Technology (ICCT), 2011, pp. 843-848.

[3] E. G. Mora, C. Greco, B. Pesquet-Popescu, M. Cagnazzo, and J. Farah. Cedar, "An optimized network-aware solution for P2Pvideo multicast," Proceedings of the 19th International Conference on Telecommunications (ICT), 2012, pp. 1-6.

[4] Y. Hongyun, H. Ruiming, C. Jun, and C. Xuhui, "A Review of Resilient Approaches to Peer-to-Peer Overlay Multicast for Media Streaming," Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2008, pp. 1-4.

[5] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan, "Resilient Multicast Using Overlays," IEEE/ACM Transactions on Networking, 2006, vol. 14, no. 2, pp. 237-248.

[6] S. Birrer and F. E. Bustamante, "Resilience in Overlay Multicast Protocols," Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2006, pp. 363-372.

[7] K. Walkowiak and M. Przewozniczek, "Modeling and Optimization of Survivable P2P Multicasting," Computer Communications, 2011, vol. 34, Issue 12, pp. 1410-1424.

[8] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiver-Driven Mesh-based Streaming," Proccedings of 26th IEEE International Conference on Computer Communication (INFOCOM), 2007, pp. 1415-1423.

[9] M. Bienkowski, M. Korzeniowski, and F. Heide, "Dynamic load balancing in distributed hash tables," Proceedings of the 4th international conference on Peer-to-Peer Systems (IPTPS), 2005, pp. 217-225.

[10] K. Walkowiak, "Survivability of P2P Multicasting, " Proceedings of the IEEE 7th International Workshop on the Design of Reliable Communication Networks, 2009, pp. 92-99.

[11] Z. Hongli, "Modeling Internet Link Delay Based on Measurement," International Conference on Electronic Computer Technology, 2009, pp. 420-424.

[12] S. Kaune, K. Leng, A. Kovacevic, G. Tyson, and R. Steinmetz, "Modelling the Internet Delay Space Based on Geographical Locations," Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing, 2009, pp. 301-310.

[13] S. Birrer, "Addresing the Limitations of Tree-based Approaches to High-Bandwith Streaming Multicast," UMI Number: 3303707. Evanston, Illinois, June. 2008.

[14] PlanetLab Consortium, Available at: http://www.planet-lab.org. [retrieved: December, 2013].

 ISBN: 978-1-61208-318-6

A Flexible P2P Gossip-based PSO Algorithm

Marco Biazzini
INRIA – Bretagne Atlantique
Rennes, France
Marco.Biazzini@inria.fr

***Abstract*—It is becoming more and more interesting, in the domain of distributed function optimization, the study of fully decentralized optimization algorithms, deployed on large networks of heterogeneous computational units. Several issues arise on such a system design, among which the proper way of distributing and making use of shared information, in absence of a centralized coordination, is a prominent one. We introduce the design of a P2P gossip-based Particle Swarm Optimization (PSO) algorithm, that is capable to implement different policies with respect to the use of global information, as this becomes available via gossiping during the computation. Such a PSO flavor is easy to tune, in order to implement different strategies, while balancing exploration and exploitation. Preliminary experimental results are shown to assess the usefulness of the proposal.**

Keywords-P2P function optimization; distributed Particle Swarm Optimization; P2P distributed computation.

I. INTRODUCTION

Distributed function optimization has a long research history [1]. Usual applicative scenarios assume the availability of either a dedicated parallel computing facility, or specialized clusters of machines. In both cases, the coordination of the distributed task is modeled in a centralized fashion, greatly simplifying its management. The limitations shown by the scalability and the robustness of these approaches are well known.

Recently, researchers have paid increasing attention to systems organized in decentralized P2P networks of solvers, distributed on a large collection of loosely-coupled machines, that cooperate to solve a common task [2], [3]. The long term goal of this kind of studies is to come up with an algorithmic design that can provide reliably good results in unsupervised and possibly heterogeneous systems.

The common requirement in these cases is that the optimization tasks must be successfully and effectively performed without any specialized infrastructure or central coordinating server being required. Ideally, these systems should self-organize themselves in a completely decentralized way, avoiding single points of failure and performance bottlenecks. The advantages of such approach are thus extreme robustness and scalability, plus the capability of exploiting existing (unused or underused) resources, like idle computer labs within a given organization, or a volunteer computing system architecture.

A reasonable approach is to partition the optimization job in a pool of independent tasks to be performed, and assign them to the available nodes, taking care of balancing the load. This can be done either using a centralized scheduler, or using a decentralized approach. This multi-algorithm approach is well known and widely used and it can be achieved in a decentralized fashion as well [4]. Anyway, it kind of "smoothes down" the challenge of finding a distributed algorithmic design, in that it uses each interconnected machine as a separate solver, rather than finding a proper decentralized design for a given algorithm to enable the cooperation of several, possibly very numerous resources.

An interesting research trend investigates a P2P approach, where a distributed algorithm spreads the load of a *single* optimization task among a group of nodes, in a robust, decentralized and scalable way [5]. By offering such a possibility, the need of solving an optimization task in a bounded time and/or with a given precision could be achieved by easily deploying identical solvers in a large-scale network and either focusing on the quality (to obtain a more accurate result by a specific deadline) or on the speed-up (to perform a predefined amount of computation over a function in the shortest possible time).

In this paper we present a novel Particle Swarm Optimization (PSO) design, that can exploit such a distributed environment and maximize the impact of a gossip-based information sharing mechanism, to avoid getting stuck in suboptimal region of the problem domain. Experiments in a real deployment show the viability of the approach and the effectiveness of the design.

In the following, Section II presents a brief description of the standard PSO algorithm and discuss some of its distributed variants. Section III outlines the distributed PSO algorithm we devise and details the novelty of its design. Section IV characterizes the distributed scenario we target and implement in our experiments. Then experimental results are presented and discussed in Section V. We draw our conclusion in Section VI.

II. BACKGROUND

We provide in this section a brief description of the standard PSO algorithm. We also recall some recent works on PSO in fully decentralized systems, to better contextualize our contribution.

A. Particle Swarm Optimization

PSO [6] is a nature-inspired method for finding global optima of functions of continuous variables. The search is performed iteratively updating a small number N (usually in the tens) of random "particles" (solvers), whose status information includes the current position vector $\boldsymbol{x}_i$, the current speed vector $\boldsymbol{v}_i$, the optimum point $\boldsymbol{p}_i$ and the *fitness* value $f(\boldsymbol{p}_i)$, which is the "best" solution the particle has achieved so far. The particle swarm optimizer also tracks the global best

 ISBN: 978-1-61208-318-6

position $\boldsymbol{g}$, in which the swarm has achieved the best fitness value obtained so far by any particle in the population.

At each iteration, every particle updates its velocity and position as described by the following equations:

$$\boldsymbol{v}_i = \boldsymbol{v}_i + c_1 \cdot rand() \cdot (\boldsymbol{p}_i - \boldsymbol{x}_i) + c_2 \cdot rand() \cdot (\boldsymbol{g} - \boldsymbol{x}_i) \quad (1)$$

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \boldsymbol{v}_i \quad (2)$$

In these equations, $rand()$ is a random number in the range $[0, 1]$, while c_1 and c_2 are learning factors, whose default values are conventionally set as $c_1 = c_2 = 2$. The pseudo code of the procedure is given in Algorithm 1.

```
foreach particle i do
  | Initialize i;
end
while maximum iterations or
      minimum error criteria is not attained do
  | foreach particle i do
  |   | Compute current fitness value f(x_i);
  |   | if f(x_i) is better than f(p_i) then
  |   |   | p_i ← x_i;
  |   | end
  | end
  | g ← bestOf(p_i), i = 1 to N;
  | foreach particle i do
  |   | Compute velocity v_i according to equation 1;
  |   | Update position x_i according to equation 2;
  | end
end
```

Algorithm 1: The standard PSO algorithm.

Particle speeds on each dimension are bounded to a maximum velocity $vmax_i$, specified by the user.

B. PSO on incomplete topologies

As one of the most investigated heuristics inspired from nature, PSO is the subject of study of a vast scientific production [7]. Given the remarkable behavioral diversity that can be obtained by tuning its parameters and shaping the way swarms interact with each others, numerous distributed variants have been brought up as well [8], [9], [10].

The above-described version of PSO assumes that all particles agree on the global best point found so far, and is often referred to as the "classical" or "full-information" version. Effects of incomplete topologies on the performance of PSO have been studied for different types of graphs [11]. Such studies were motivated by the observation that incomplete topologies may prevent the system from concentrating too much on early-found local optima, therefore improving solution quality. Full information has generally been shown to outperform partial topologies [12]. Yet, our work focuses on cases where incomplete information is a consequence of the network topology, and global data maintenance is not practical. Some recent publications presented PSO flavors adapted for P2P overlay networks [13], [14].

Improvements with respect to naiver versions and robustness even in faulty environments have been shown [15], [16], that rely mostly on the periodic diffusion of the current best solution among the distributed solvers and exploit the effectiveness of gossip protocols in spreading relevant information among peers. In general, gossip-based distributed computing has shown to be able to drive the gradual improvement of evolutionary algorithms, while achieving both scalability and quality [17].

Within this context, it is still poorly understood how to optimally use the information that is gossiped from node to node. We argue that is not only the rate of gossiping that affects the performance, as it has been shown [18]. Once we have the most up-to-date information available at each peer, the local solver still can choose whether to use it as soon as possible, or to schedule the utilization in a strategic way. The contribution of this paper is to present the design of a P2P gossip-based PSO algorithm that is capable to implement different policies with respect of the use of global information. Differing from the works cited above, our approach focus on the way information generated remotely is handled locally at each solver, rather than on studying the performance of the information spreading protocol, some global population control mechanism or the optimal setting of the basic PSO parameters.

III. Algorithm Description

The distributed PSO algorithm we propose offers a novel way to improve the exploration of the search domain, not to cut the search short towards the current best solution (likely to be suboptimal). The idea is to have an algorithm that can choose among different policies. These policies determine how and when the global information about the current best value found — available at any time as communicated by the other peers — should be used. At least two good reasons not to immediately consume the shared data come to mind:

1) to mitigate the event of an early convergence to suboptimal attraction basins, by not moving too fast towards the current optimal value;
2) to enhance the exploration of the search space, by avoiding that the swarms get too close to each other at an early stage.

By applying policies that define how to use the knowledge about the current global optimum, the PSO algorithm can be tuned in a way that is easier for the user to understand, with respect to the tuning of the various PSO parameters, whose behavioral effects are often obscure or at least debatable. We are aware that top quality results on hard problems can only be achieved by a careful and clever tuning of the algorithm on function characteristics. Our contribution and the results of our experiments point out, nonetheless, that attention should be paid not only to the core algorithmic parameters, but also to the way the shared information is diffused and consumed by the various agents of the distributed optimization task.

The key idea of our proposal is to maintain an ever-refreshing knowledge of the best point evaluated so far by any swarms in the network, but without necessarily substituting the local swarm's global best with this value. The overall global best should be rather used at a time and in a way that serves a given strategy. Table I gives the description of the notation we use in the following.

 ISBN: 978-1-61208-318-6

TABLE I: Description of the adopted notation.

Notation	*Meaning*
gb	the local swarm's global best
ogb	the overall global best, periodically gossiped among the solvers
p	the given policy to apply
E	overall number of function evaluations performed in the network

We can describe the general design of our distributed PSO algorithm as follows. Each swarm iteratively performs these steps:

1) *update* the **ogb** via a gossip message exchange
2) *apply* **p** to decide about using either **gb** or **ogb** to move the current particle
3) *move* the particle according to the decision taken
4) *evaluate* the function in the particle's position
5) *update* the records about the local and global best values (both **gb** and **ogb**) as needed

The strategic decisions that will impact the behavior of the algorithm are then implemented in the policy **p**. The policy may be simple or very complex, may use a limited amount of local knowledge or it may use any shared information (beside the value of **ogb**) that can be made available via peer-wise communication among the solvers. Trying to give a minimal set of requirements, we consider that a good policy should indicatively specify:

– how to decide to use **ogb** in a given PSO iteration (what triggers the decision, if it depends and involves the whole swarm or the single particles, etc.);
– for how many subsequent iterations **ogb** will substitute **gb** to compute the speed of the particles (how long each application of the current global optimum will last in the local solver);
– when this substitution shall permanently or temporarily end (what determines the end of each "**ogb** session").

In Section V we show experimental results obtained by running the described distributed PSO algorithm with a simple policy.

IV. Distributed Framework Characteristics

We target the general framework described in [18], thus considering a parallel islands scenario, in which several swarms of particles are initialized at random over a function domain. Each swarm is hosted by a peer and peers are distributed in a random overlay. During the search, every swarm periodically exchanges information with another swarm hosted by a peer, that is selected at random from the local neighborhood. At each peer, the neighborhood is maintained by means of a peer sampling service, implemented by the NEWSCAST gossip protocol [19].

All the communication mechanisms are based on gossip algorithms implemented on top of this service. Each peer always propagates the current best solution to others, by periodically sending it to one randomly chosen peer. Upon receiving this information, a peer updates its own current best solution, but only if the received one is better. If this is not the case, no further information is sent back to the sender. Thus the epidemic protocol implements a simple *push* approach. The period of gossip is assumed to be at least that of one function evaluation, thus a communication event is triggered after each function evaluation at all peers. As it is known by the behavior of the epidemic protocols, the time to propagate this way a new-found best solution to every node is logarithmic (in expectation) with respect to the size of the network.

This kind of lightweight and asynchronous communication among swarms suits well a large-scale, possibly heterogeneous environment. Though it could be beneficial in terms of absolute performance, the swarms are not required to perform a similar number of function evaluations in a given time, nor they have tight time constraints to perform mutual information exchanges.

TABLE II: Test functions. L: number of local minima.

	Function $f(x)$	L
Rosenbrock20	$\sum_{i=1}^{19} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	1
Zakharov20	$\sum_{i=1}^{20} x_i^2 + (\sum_{i=1}^{20} i x_i/2)^2 + (\sum_{i=1}^{20} i x_i/2)^4$	1
Rastrigin20	$200 + \sum_{i=1}^{20} [x_i^2 - 10\cos 2\pi x_i]$	$\approx 10^6$
Griewank20	$\sum_{i=1}^{20} x_i^2/4000 - \prod_{i=1}^{20} \cos\left(x_i/\sqrt{i}\right) + 1$	$\approx 10^{19}$

V. Experimental Results

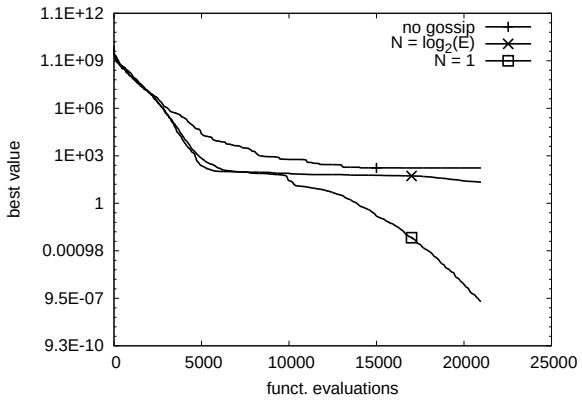
The results presented in this section have been obtained in a real distributed environment. We use our open source Java implementation of a distributed optimization framework [20] and the grid facilities provided by Grid5000 [21] In this kind of experiments we do not focus on the absolute performance of the algorithm, but rather on the differences among the adopted configurations.

We deploy 50 solvers (swarms) on an equal number of machines on the grid. Their random P2P overlay is maintained by the NEWSCAST gossip protocol, running on each machine. At each peer, the local neighborhood constantly represents a random subset of the network. At the beginning of each PSO iteration, a swarm updates its local information according to the messages that have been received since the update phase of the previous iteration. At the end of each PSO iteration, a peer solver is selected at random by each peer in a local neighborhood of 20 peers as the recipient of an update message. No churn and no faults are considered in this scenario. At each peer, the solver consists of a swarm of 4 particles, whose parameters are set as follows: $w_1 = 0.9$, $w_2 = 0.4$, $c_1 = 2$, $c_2 = 2$.

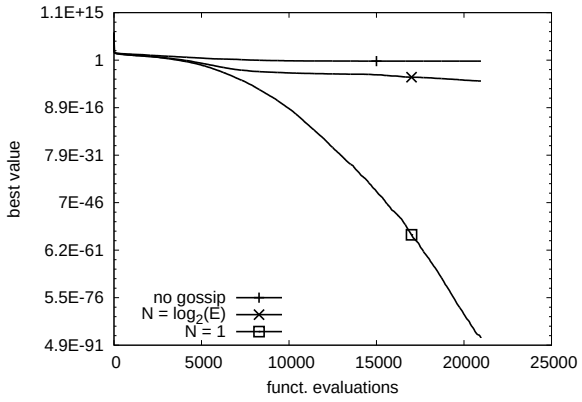
We evaluate four well known benchmark functions, described in Table II. They differ in the number and the distribution of their local minima, whereas the value of the global minimum is 0 for all of them. We perform 10 runs for each experiment, taking the average best value. The overall number of function evaluations in the network, equally partitioned among the solvers, is set as $E = 2^{20}$. The policy adopted for these experiments is the following:

– start using **ogb** instead of **gb** whenever **gb** has not been improved in the latest N iterations;

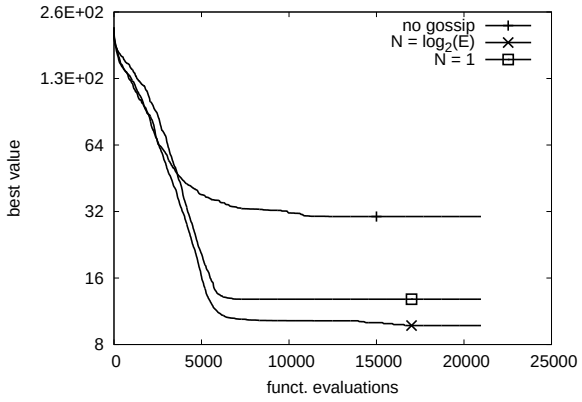
 ISBN: 978-1-61208-318-6



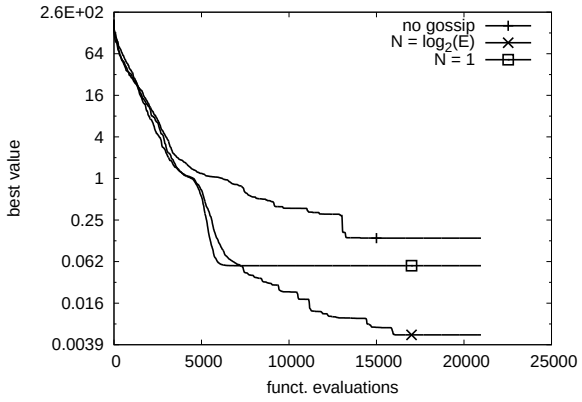
(a) Rosenbrock20



(b) Zakharov20



(c) Rastrigin20



(d) Griewank20

Fig. 1: Results on unimodal and multimodal test functions.

– continue using **ogb** until each particle in the swarm has been moved at least once *AND* **gb** is not improved (this being both a duration and a stopping criterion).

For each function, we run different experiments by choosing $N = log_2(E)$ and $N = 1$. This latter value means that **ogb** is always used instead of **gb**, thus making PSO behaving like the version presented in previous works [18], [15]. We run one more set of experiments with no data exchange among the peers (thus making **ogb** of no use), to compare the performance.

As Figure 1 shows, our design allows to implement a flexible PSO algorithm, whose performance varies according to the strategy implemented by the given policy. The results we show for the Rosenbrock function (Figure 1(a)) seem to confirm the idea that the faster the gossiped information is spread and used among the solvers, the better will be the final result. It can be clearly seen how the outcomes get increasingly better while we choose to use **ogb** more and more often, with respect to **gb**. The same trend is visible in the outcomes or the Zakharov function (Figure 1(b)), where the distributed PSO algorithm is known to perform really well. The logarithmic scale of the vertical axis emphasizes the effect of the full exploitation of **ogb**, but we can see that by applying our policy we are anyway able to improve the baseline by some orders of magnitude. We notice that both the Zakharov and the Rosenbrock functions are unimodal.

The case for the Rastrigin function (Figure 1(c)) and for the Griewank function (Figure 1(d)) is quite different. It turns out that, by slowing down the rate at which PSO prefer **ogb** over **gb** throughout the computation, we can actually avoid an early convergence to suboptimal basins. The results with the Rastrigin function are particularly significant, because PSO is known to have severe troubles in getting out from its local minima. Thus, by implementing a simple policy about the usage of the available global knowledge, we are able to obtain improvements that would otherwise cost a long time spent in tuning the basic PSO parameters towards a "good" configuration. We notice that both the Rastrigin and the Griewank functions are multimodal.

As a general remark, our design seems to improve PSO's ability of escaping from local minima, whereas it slows down the pursuit of the global optimum when PSO is searching smoother domains. Thus, we may conclude that using the proposed policy is useful while optimizing functions that are known (or expected) to present several local (suboptimal) attractors.

Furthermore, we notice that the policy implemented in our experiment is very simple and static. This can be the main reason why, being anyway capable to improve the solution quality for multimodal functions, it is not effective enough to prevent PSO to be eventually trapped to suboptimal basins, as the long horizontal lines of Figures 1(c) and 1(d) clearly show. The same reason could be behind the analogous phenomenon of excessively slow improvement showed by Figures 1(a) and 1(b). A policy that dynamically adapts to the current state of the computation, as new information becomes available to each peer via usual decentralized mechanisms, can lead to better results and is currently being investigated.

 ISBN: 978-1-61208-318-6

The limited set of results obtained so far is not enough to assess a generalized pattern that holds in different scenarios. Nonetheless, we believe it may point out a novel profitable research direction, which have a large potential to be deepened and extended. Particularly interesting may be the analysis of the behavior of the algorithm with respect to different values of N, or with respect to different local swarms sizes. Both of these parameters can have a significant impact on the performance of the algorithm, which we intend to examine in our future work.

VI. Conclusion

The contribution of this paper belongs to the emerging domain of P2P-distributed function optimization. It is particularly important in such a domain, besides the tuning of the optimization algorithm itself, the way useful information is shared among the participants and how each of them chooses to use it.

We presented the design of a P2P gossip-based PSO algorithm that is capable to implement different policies with respect to the use of gossiped information about the overall best point known at any time in the network. Preliminary experimental results show how the performance of such an algorithm can vary, depending on how quickly each solver makes use of the information sent by other peers. The outcomes show that the quality of the optimization can benefit from adopting flexible strategies like the one we propose. This may lead to a better exploration of the function domain and help avoiding early suboptimal convergence.

Our results, as those of previous works [5], [16] on P2P decentralized function optimization, show that exploiting large scale, loosely coupled and possibly heterogeneous distributed systems to obtain good quality results is a viable approach. Particular care must be dedicated not only to modify the algorithms to fit the specific distributed environment, but also to model the diffusion of information among the participants in the most effective way. Among the other possible research directions, we think that is of utmost interest the study of how different overlay communication topologies and different computational capacities of the peers may affect the overall quality of the results.

Acknowledgements

Experiments presented in this paper were carried out using the Grid5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

This work was partially funded by Bretagne regional project SOFTLIVE.

References

[1] E. G. Talbi, *Parallel Combinatorial Optimization*. John Wiley and Sons, USA, 2006.

[2] J. Laredo, P. Castillo, A. Mora, and J. Merelo, "Exploring population structures for locally concurrent and massively parallel evolutionary algorithms," in *Evolutionary Computation, 2008. CEC 2008.*, June 2008, pp. 2605–2612.

[3] N. Melab, M. Mezmaz, and E.-G. Talbi, "Parallel hybrid multi-objective island model in peer-to-peer environment," in *Proceedings of IPDPS'05*. Washington, DC, USA: IEEE Computer Society, 2005, p. 190.2.

[4] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Transactions on Computer Systems*, vol. 23, no. 3, pp. 219–252, August 2005. [Online]. Available: cikkek/tocs04.pdf

[5] J. Laredo, P. Castillo, A. Mora, J. Merelo, and C. Fernandes, "Resilience to churn of a peer-to-peer evolutionary algorithm," International Journal of High Performance Systems Architecture, 2008, volume 1, Number 4.

[6] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *IEEE Int. Conf. Neural Networks*, pp. 1942–1948, 1995.

[7] R. Poli, "Analysis of the publications on the applications of particle swarm optimisation," Journal of Artificial Evolution and Applications, November 2007.

[8] U. K. Wickramasinghe and X. Li, "Using a distance metric to guide pso algorithms for many-objective optimization," in *Proceedings of the 11th Genetic and Evolutionary Computation Conference (GECCO'09)*, Montreal, Québec, Canada, Jul. 2009, CONFERENCE, pp. 1339–1346.

[9] T. Desell, M. Magdon-Ismail, B. Szymanski, C. Varela, H. Newberg, and D. Anderson, "Validating evolutionary algorithms on volunteer computing grids," in *Distributed Applications and Interoperable Systems*, ser. Lecture Notes in Computer Science, F. Eliassen and R. Kapitza, Eds. Springer Berlin / Heidelberg, 2010, vol. 6115, pp. 29–41. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13645-0_3

[10] M. R. Khouadjia, L. Jourdan, and E.-G. Talbi, "Adaptive Particle Swarm for Solving the Dynamic Vehicle Routing Problem," in *In Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)*, Tunisie, Sep. 2010, p. 10.1109/AICCSA.2010.5587049. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00525446/en/

[11] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proc. 4^{th} Congress on Evolutionary Computation (CEC'02)*, May 2002, pp. 1671–1676.

[12] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: Simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, Jun. 2004.

[13] I. Scriven, A. Lewis, D. Ireland, , and J. Lu, "Distributed multiple objective particle swarm optimisation using peer to peer networks," in *IEEE Congress on Evolutionary Computation (CEC)*, 2008.

[14] I. Scriven, A. Lewis, and S. Mostaghim, "Dynamic search initialisation strategies for multi-objective optimisation in peer-to-peer networks," *IEEE Congress on Evolutionary Computation, CEC '09*, pp. 1515 – 1522, 2009.

[15] M. Biazzini, B. Bánhelyi, A. Montresor, and M. Jelasity, "Peer-to-peer optimization in large unreliable networks with branch-and-bound and particle swarms," in *Applications of Evolutionary Computing*, Mario Giacobini *et alii*, Ed. Springer, 2009, pp. 87–92.

[16] M. Biazzini and A. Montresor, "Gossiping differential evolution: a decentralized heuristic for function optimization in p2p networks," in *Proceedings of the 16th International Conference on Parallel and Distributed Systems (ICPADS'10)*, Dec. 2010.

[17] J. L. J. Laredo, E. A. Eiben, M. Schoenauer, P. A. Castillo, A. M. Mora, F. Fernandez, and J. J. Merelo, "Self-adaptive gossip policies for distributed population-based algorithms," 2007. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0703117

[18] M. Biazzini, A. Montresor, and M. Brunato, "Towards a decentralized architecture for optimization," in *Proc. of IPDPS'08*, Miami, FL, USA, April 2008.

[19] M. Jelasity, S. Voulgaris, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "Gossip-based peer sampling," *ACM Transactions on Computer Systems*, vol. 25, no. 3, p. 8, August 2007.

[20] M. Biazzini and A. Montresor, "P2poem: Function optimization in p2p networks," *Peer-to-Peer Networking and Applications*, pp. 1–20, 10.1007/s12083-012-0152-8. [Online]. Available: http://dx.doi.org/10.1007/s12083-012-0152-8

[21] https://www.grid5000.fr.

 ISBN: 978-1-61208-318-6

Simulation of Buffering Mechanism for Peer-to-Peer Live Streaming Network with Collisions and Playback Lags

Yuliya Gaidamaka, Ivan Vasiliev,
Andrey Samuylov, Konstantin Samouylov
Telecommunication Systems Department
Peoples' Friendship University of Russia
Moscow, Russia
E-mail: ygaidamaka@sci.pfu.edu.ru, iuvasiliev@gmail.com,
asam1988@gmail.com, ksam@sci.pfu.edu.ru

Sergey Shorgin
The Institute of Informatics Problems
The Russian Academy of Sciences
Moscow, Russia
E-mail: sshorgin@ipiran.ru

***Abstract*—In this paper, an approach to the peer-to-peer live streaming network simulation is presented. The model considers collisions because of limitation of peer's upload capability and takes into account the data transfer delays, which cause playback lags between peers. As a basis of simulation the mathematical model is considered, which describes in terms of discrete Markov chain the data exchange process between buffers of users in peer-to-peer network. Joint analysis of the two models — simulation and mathematical — leads to better understanding the impact of collisions and playback lags on playback continuity which is necessary when designing the effective peer-to-peer live streaming network.**

Keywords—P2P live streaming network; buffer occupancy; playback continuity; Markov chain model; playback lags

I. INTRODUCTION

Peer-to-peer **(**P2P) network is a kind of overlay content delivery network which consists of users who make their resources (computing power, memory, and bandwidth) available to other users without central coordination. In P2P networks, users not only download data, but also simultaneously distribute the downloaded data to other users, thus, peer-to-peer networking differs from client-server networking.

There are two types of P2P networks: file-sharing and streaming P2P networks [1][2]. In both cases, users download content as small blocks of data called chunks and each user downloads the missing chunks from other users, who have already downloaded them. In file-sharing networks (known as BitTorrent-like networks), users have to download the entire file before they begin to use it, so that a user is not restricted by time to obtain any chunk. In streaming networks, users simultaneously download and play the video stream, so a limit for download time of a chunk is crucial, since every chunk has its playback deadline. To provide smooth playback in streaming P2P networks the buffering mechanism is utilized. Each user has a buffer for caching the most recently downloaded data chunks. Moreover, only the chunks that are yet to be played will be downloaded. In both cases, in order to select which chunk to download next, a download strategy, such as Rarest First, Latest Useful Chunk First (LF), Greedy, Rarest Random, Naive Sequential, Cascading, and Hybrid download strategies, is applied [3][4].

Peer-to-peer network performance measures are usually analyzed via using different mathematical models. The so called fluid models are used to analyze file sharing networks [5-8]. One of the main performance metrics of file-sharing networks is how long it takes to download the whole file (the file download time or latency). Streaming P2P networks are stricter in respect to performance measures, and their distinctive feature is that they are generally analyzed in discreet time [9-16] with much attention paid to investigation of the buffering mechanism [11-16]. In streaming networks, the main performance measures are the startup delay (or latency), playback continuity (or skip-free playout probability) and the probability of universal streaming.

In this paper, a simulation model for analyzing the data exchange process in P2P live streaming network is presented. Like the model of [16], our model is built on the scheme of chunk exchange between peers' buffers introduced in [11]. The model takes into account limitation of peer's upload capability the result of which becomes violation of playback smoothness. The corresponding performance measure is probability of playback continuity, one of Quality of Experience (QoE) parameters in P2P networks. Unlike [11] and [16], our model also considers the playback lags between peers, which were first discussed in [12]. As a basis of simulation the mathematical model is considered, which describes in terms of discrete Markov chain the data exchange process between buffers of users in P2P live streaming network. The mathematical model is based on the model introduced in our previous work [14] and was modified for LF download strategy in order to take into account collisions and playback lags. Joint analysis of the two models — simulation and mathematical — provides advantages in the development of the algorithm for modeling and allows to improve accuracy of the calculations. Two main characteristics were investigated as a function of peer's upload capability — the probability of collision and the probability of playback continuity. By numerical example, the impact of collisions and playback lags on playback continuity is illustrated. Our main results are the following.

- The rigorous mathematical model of the download strategy in terms of a discrete Markov chain, that takes into account collisions and playback lags unlike [11] and [16].
- The exact formula for the index of peer's buffer position to download a chunk according to LF

 ISBN: 978-1-61208-318-6

download strategy considering the playback lags between peers unlike [13] and [14].
- The detailed algorithm of chunk exchange between buffers of peers considering collisions unlike [13] and [14].
- The exact formula for the probability of playback continuity considering the playback lags unlike [11-16].

This paper is organized as follows. In Section II, a video data distribution in a P2P live streaming network with buffering mechanism is described, and the mathematical model of the download strategy, in terms of a discrete Markov chain with a rigorous mathematical description, is considered. Also, the detailed algorithm of chunk exchange between buffers of peers in P2P live streaming network is developed and main performance measures are defined. In Section III, performance analysis and some case study is performed. The conclusion of this paper is presented in Section IV.

II. MODEL

In this section, a video data distribution in a P2P live streaming network with buffering mechanism is studied. Consider a P2P network with N users present in the network, and a single server, which transmit only one video stream. The process of video stream playback is divided into time slots, the length of each time slot corresponds to the playback time of one chunk. Each user has a buffer designed to accommodate M+1 chunks, where the buffer positions are numbered from 0 to M: 0-position is to store the freshest chunk just received from the server, other m -positions, $m = 1,...,\mathrm{M}-1$, are to store chunks, already received during the past time slots or will be downloaded in the coming time slots, and buffer M-position is to store the oldest chunk that will be moved out from the buffer for playback during the next time slot.

Let us specify the actions that the server and users perform during each time slot. At the beginning of each time slot the server randomly selects a user from the network and uploads the newest chunk into his buffer 0-position. Any other user, not chosen by the server during the current time slot, will perform the following actions. If there are empty positions in the user's buffer (i.e., there are missing chunks in his buffer) the user will randomly choose another user, called a target user, from the predefined group of his neighbors in order to download one of the missing chunks from him. The number of chunks that a target user can upload is restricted by its upload capability. So, if the number of users that chose the same user as a target user exceeds the target user upload capability then a collision occurs. In case of collision, the number of users that successfully download missing chunks corresponds to the target user upload capability and the others don't download anything at all. If no collision occurs and the target user has one of the missing chunks, then the attempt to download from the target user will be successful. If the target user has more than one of the missing chunks, then download strategy will define which chunk to download. One of the simplest used strategies is LF strategy. With the LF strategy during any time slot each user tries to download the appropriate chunk with minimum index [12]. A user will not download any chunk in the current time slot at all, if in the current time slot all positions of his buffer are occupied (there are no empty positions) or if the target user he have chosen does not have any of the missing chunks. At the end of each time slot, chunks in the buffer of each user shift one step forward, i.e., the chunk in M-position moves to the player for playback, the buffer 0-position gets free to accommodate a new chunk from the server at the beginning of the next time slot. The remaining chunks in other positions shift one position to the right (towards the end of the buffer) to replace the position freed by its predecessor.

Below a mathematical model for chunk exchange between user's buffers is developed in the form of discrete Markov chain. The model of user behavior, proposed in [13][15], is extended by taking into account data transfer delays called playback lags that affect the video data exchange process between users as it is shown in Figure 1.

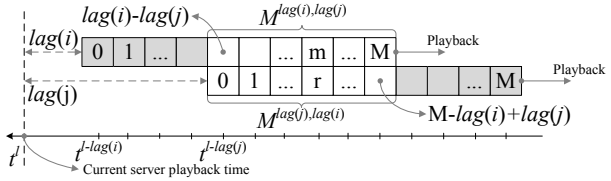


Figure 1. Buffers states mapping with playback lags

For a given network with N users and the single server, vector $\mathbf{z}(n) = \left(lag(n), u(n), \mathbf{x}(n)\right)$ defines the state of each user (n-user), where $lag(n)$ is the data transfer delay from server (playback lag), $u(n)$ is n-user upload capability and $\mathbf{x}(n) = \left(x_0(n), x_1(n), \cdots, x_{\mathrm{M}}(n)\right)$ is the state of n-user's buffer. Here $x_m(n)$ is the state of n-user's buffer m-position: $x_m(n) = 1$, if n-user's buffer m-position is occupied with a chunk, otherwise $x_m(n) = 0$, where m is the index of position in user's buffer, $m \in \{0,1,...,\mathrm{M}\}$. Each user in the network uses buffer positions $m = 1,...,\mathrm{M}$ to store the chunks downloaded from the other users, and uses 0-position only to store the chunk downloaded from the server. Note that, if during any time slot M-position is occupied, then n-user watches the video stream without any pause.

Thus, the state of the system is defined by $\mathbf{Z} = (\mathbf{lag}, \mathbf{u}, \mathbf{X})$, where $\mathbf{lag} = \left(lag(1),...,lag(N)\right)$ and $\mathbf{u} = \left(u(1),...,u(N)\right)$ are vectors that define the playback lag, and the upload capability for each user, and the n-th row of the matrix $\mathbf{X}$ corresponds to the buffer state of n-user, $\dim \mathbf{X} = \mathrm{N}(\mathrm{M}+1)$.

Denote by $M^0\left(\mathbf{x}(n)\right)$ and $M^1\left(\mathbf{x}(n)\right)$ the set of indexes of all empty (1) and occupied (2) positions in n-user's buffer respectively:

$$M^0\left(\mathbf{x}(n)\right)=\left\{m:\ x_m(n)=0,\ m=1,...,\mathrm{M}\right\}, \quad (1)$$

$$M^1\left(\mathbf{x}(n)\right)=\left\{m:\ x_m(n)=1,\ m=1,...,\mathrm{M}\right\}. \quad (2)$$

Here $M^0\left(\mathbf{x}(n)\right)\subseteq\{1,...,\mathrm{M}\}$, $M^1\left(\mathbf{x}(n)\right)\subseteq\{1,...,\mathrm{M}\}$, and $M^0\left(\mathbf{x}(n)\right)\bigcup M^1\left(\mathbf{x}(n)\right)=\{1,...,\mathrm{M}\}$ is the set of indexes of all positions in n-user's buffer available for download from other user, not from the server. Note that due to playback lags not the entire buffer of a user is available for chunk exchange, see Figure 1. As in (3) for arbitrary i-user and j-user, $i,j\in\{1,...,\mathrm{N}\}$, the set $M^{lag(i),lag(j)}$ determines the indexes of i-user buffer positions that are available for chunk exchange with j-user:

$$M^{lag(i),lag(j)}=\begin{cases}\{0,1,\ldots,\mathrm{M}-lag(i)+lag(j)\}, & \text{if } lag(i)\geq lag(j),\\ \{lag(i)-lag(j),\ldots,\mathrm{M}\}, & \text{if } lag(i)<lag(j).\end{cases} \quad (3)$$

Then, for n-user and h-user the intersection $M^0\left(\mathbf{x}(n)\right)\bigcap M^{lag(n),lag(h)}$ is the set of the indexes of n-user's empty buffer positions, one of which could be filled in with data from h-user. And the intersection $M^1\left(\mathbf{x}(h)\right)\bigcap M^{lag(h),lag(n)}$ is the set of the indexes of occupied positions in h-user's buffer, that h-user can upload corresponding chunk to n-user.

Due to data transfer delays, one and the same data chunk in the buffers of users with different playback lags is located in positions with different indexes. In order to establish a correspondence between these positions, the following operation is used: $m=r-lag(n)+lag(h)$. Here m is an index of buffer position for n-user, and r is a corresponding index of buffer position for h-user, $m\in M^{lag(n),lag(h)}$, $r\in M^{lag(h),lag(n)}$. Thereby, the index $m_{LF}\left(\mathbf{x}(n),\mathbf{x}(h),lag(n),lag(h)\right)$ of n-user's buffer position to which n-user according to LF download strategy should try to download a chunk from h-user is determined by the following formula:

$$\begin{aligned}&m_{LF}\left(\mathbf{x}(n),\mathbf{x}(h),lag(n),lag(h)\right)=\\&=\min\left\{\left(M^0\left(\mathbf{x}(n)\right)\bigcap M^{lag(n),lag(h)}\right)\bigcap\right.\\&\left.\bigcap\left\{\ m:m=r-lag(n)+lag(h),\right.\right.\\&\left.\left.r\in\left(M^1\left(\mathbf{x}(h)\right)\bigcap M^{lag(h),lag(n)}\right)\right\}\right\}.\end{aligned} \quad (4)$$

Denote by $S\mathbf{x}(n)$ the shifting operator of vector $\mathbf{x}(n)$, meaning if $\mathbf{x}(n)=\left(x_0(n),x_1(n),\cdots,x_{\mathrm{M}-1}(n),x_{\mathrm{M}}(n)\right)$, then $S\mathbf{x}(n)=\left(0,x_0(n),\cdots,x_{\mathrm{M}-1}(n)\right)$. Let t_l be the shifting instant of buffer contents. When constructing the model in a discrete time, it is assumed that if at the instant t_l-0 a buffer is in the state $\mathbf{x}(n)$, then at the instant t_l+0 it will be in the state $S\mathbf{x}(n)$.

According to the protocol for the data distribution in P2P live streaming network with a buffering mechanism, in the interval $\left[t_l,t_{l+1}\right)$, which corresponds to the l-th time slot, the server and users perform the following actions.

1) At the instant t_l for all users the shift of the buffer content takes place:

a) Chunk in buffer M-position if present will be sent for playback;

b) All other chunks in other buffer positions will be shifted one position to the right, i.e., towards the end of the buffer;

c) Buffer 0-position will be emptied.

2) At the instant t_l+0 server chooses one user randomly and uploads a chunk for the current time slot to his buffer 0-position. If server has chosen i-user, then $x_0(i)=1$ at the instant $t_{l+1}-0$.

3) Each user (n-user), not chosen by the server, randomly chooses one of his neighbors (h-user). Let $C^l(h)$ be the number of users, which chose h-user as a target user at the l-th time slot.

a) If $C^l(h)\leq u(h)$ (case "no collision") then n-user tries to download one of the missing chunks from h-user in its buffer's $m_{LF}\left(\mathbf{x}(n),\mathbf{x}(h),lag(n),lag(h)\right)$ position according to LF download strategy.

b) If $C^l(h)>u(h)$ (case "collision") then h-user chooses $u(h)$ users from $C^l(h)$ users randomly and each of chosen users tries to download one of the missing chunks from h-user in its buffer's $m_{LF}\left(\mathbf{x}(n),\mathbf{x}(h),lag(n),lag(h)\right)$ position according to LF download strategy. The other $C^l(h)-u(h)$ users go flop with downloading during the l-th time slot.

Denote by $\mathbf{Z}^l=\left(\mathbf{lag},\mathbf{u},\mathbf{X}^l\right)$ the network state at the instant t_l-0 and then the set $\left\{\mathbf{Z}^l\right\}:=\left\{\mathbf{Z}^l,\ l\geq 0\right\}$ forms a Markov chain over state space Ω with one class $\tilde{\Omega}$ of essential states, $\tilde{\Omega}\subset\Omega$. Let $\pi^l(\mathbf{Z})$ be the probability that Markov chain $\left\{\mathbf{Z}^l\right\}$ during l-th time slot is in state $\mathbf{Z}$, i.e., $\pi^l(\mathbf{Z})=P\{\mathbf{Z}^l=\mathbf{Z}\}$, $\mathbf{Z}\in\Omega$. The probability distribution $\pi^l(\mathbf{Z})$ is obtained in [14], the analytical formulas for calculating transition probability matrix of Markov chain $\left\{\mathbf{Z}^l\right\}$ is obtained in [15].

 ISBN: 978-1-61208-318-6

III. PERFORMANCE ANALYSIS AND SOME CASE STUDY

One of the main performance measures of P2P live streaming network is the probability $PV(n)$ of playback continuity, which is the probability that buffer M-position of n -user is occupied with the corresponding chunk for playback by the end of any time slot. To find this probability, the function

$$H_n^m(\mathbf{Z}) = \sum_{\substack{h=1,...,N,\\ h\neq n}} \delta_{m_{LF}(\mathbf{x}(n),\mathbf{x}(h),lag(n),lag(h)),m}, \quad (5)$$

$m = 1,...,\mathrm{M}$, is introduced. The function $H_n^m(\mathbf{Z})$ corresponds to the number of h -users who have a chunk in their buffer m -position, from which n -user can download in accordance with the LF download strategy when the network is in the state $\mathbf{Z} \in \Omega$. Here

$$\delta_{i,j} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \quad (6)$$

Now, the probability $Q_n^l(m)$ that during the l-th time slot the chunk which n -user can download to his buffer m-position is available in the network is defined. Due to the dependency of this probability on the downloading strategy the function $Q_n^l(m)$ can be interpreted as the probability that n-user will select m-position and successfully download a chunk from the target user during the l-th time slot. If $N \geq 2$, then one can obtain the following formula:

$$Q_n^l(0) = 0,$$

$$Q_n^l(m) = \frac{1}{N-1} \sum_{\mathbf{Z}\in\Omega} \pi^l(\mathbf{Z}) \cdot H_n^m(\mathbf{Z}),\ m = 1,...,\mathrm{M}. \quad (7)$$

Denote by $p_0^l(n,m)$ ($p_1^l(n,m)$) the probability that m-position of n-buffer is empty (occupied) during l-th time slot. Then, we can obtain a recursive relation for calculating the buffer state probabilities in a following form:

$$p_1^l(n,0) = 1/N, \quad (8)$$

$$p_1^{l+1}(n,m+1) = p_1^l(n,m) + p_0^l(n,m) Q_n^l(m),\ m = 0,...,M-1.$$

Assume that the equilibrium distribution of the Markov chain $\{\mathbf{Z}^l\}$ exists. Denote by $p_1(n,m) = \lim_{l\to\infty} p_1^l(n,m)$ the probability that m-position of n-buffer is occupied and by $p_0(n,m) = \lim_{l\to\infty} p_0^l(n,m)$ the probability that m-position of n-buffer is empty, and $Q_n(m) = \lim_{l\to\infty} Q_n^l(m)$. Then, the following equation can be obtained:

$$p_1(n,0) = 1/N,$$

$$p_1(n,m+1) = p_1(n,m) + p_0(n,m) Q_n(m),\ m = 0,...,M-1. \quad (9)$$

Thus, the probability $PV(n)$ that n-user is watching video without pauses during playback, i.e., the probability of playback continuity, is defined by the following formula:

$$PV(n) = p_1(n,M). \quad (10)$$

Let us denote by $PC(h)$ the probability of collision for h -user, i.e., the situation when the number $C(h)$ of users that chose h -user as a target user exceeds the value of upload capability $u(h)$ of h-user. Thus, the formula is obtained:

$$PC(h) = \lim_{l\to\infty} P\{C^l(h) > u(h)\},\ h = 1,...,N. \quad (11)$$

On the basis of the above results, a simulator was developed for analysis of a P2P live streaming network with following values of parameters: N=300, M=40, and the number of neighbors is equal to 60. It is assumed that all users have the same upload capability $u(h) = u, h = 1,...,N.$ Therefore $PC(h) = PC,\ h = 1,...,N.$

As it is shown in (12), the set of all users is split into three equal-sized non-overlapping groups for simplicity, assuming that the playback lags for all users in one group are the same, i.e.,

$$N = \bigcup_{k=1}^{3} N_k,\ lag(n) = lag(n'),$$
$$n, n' \in N_k,\ k = 1,2,3. \quad (12)$$

The playback lag of the first group is set to zero and the playback lag of the second and third groups are 10 and 20 time slots respectively.

Then, the simulation was conducted according to the algorithm described in Section II. The simulation runs for a certain amount of simulation time equal to 1 000 000 time slots, as extending the simulation time did not affect the results. The statistics was gathered starting with the 50 000-th time slot in order to negate the non-steady state time interval.

The graphs in Figure 2 and Figure 3 show how the probability of collision and the probability of playback continuity depend on the user's upload capability. The corresponding 95% confidence intervals are not shown in the figures because of the scale. The confidence intervals are given in Table 1.

The graph in Figure 2 shows that the probability of collisions decreases with increasing the user's upload capability. The graphs in Figure 3 show that the users of the group with the largest value of the playback lag (the third group) have the greatest probability of watching video stream without pauses in playback, e.g., without freezes and reboots.

 ISBN: 978-1-61208-318-6

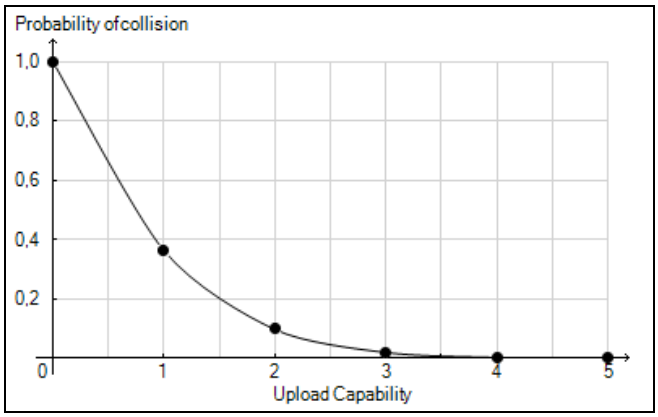


Figure 2. Probability of collision

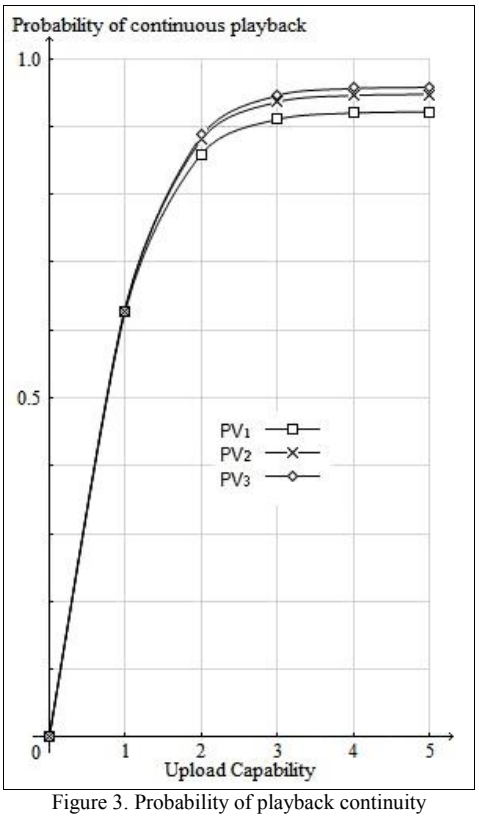


Figure 3. Probability of playback continuity

TABLE I. THE 95% CONFIDENCE INTERVALS

Upload capability	*Probability of playback continuity*			*Probability of collision*
u	PV_1	PV_2	PV_3	PC
1	[0.627228, 0.628382]	[0.631887, 0.633776]	[0.631704, 0.633594]	[0.362913, 0.364799]
2	[0.85844, 0.859804]	[0.879786, 0.881058]	[0.885195, 0.886442]	[0.097177, 0.098341]
3	[0.909756, 0.910876]	[0.935353, 0.936314]	[0.945026, 0.945916]	[0.020048, 0.020601]
4	[0.919197, 0.920262]	[0.945085, 0.945974]	[0.955367, 0.956173]	[0.00343, 0.003663]
5	[0.920827, 0.921882]	[0.946741, 0.947618]	[0.957117, 0.957907]	[0.000486, 0.000577]

The reason is that any data chunk becomes highly available among users of the first and second groups by the time it is requested by the users of the third group.

IV. CONCLUSION

In this paper, the approach to simulation of the buffering mechanism in P2P live streaming network with collisions and playback lags is presented. The advantages of the mathematical model were used to develop the simulator for the performance evaluation of the QoE parameters including the probability of collision and the probability of playback continuity.

The direction of future research is simulation and comparison of most popular download strategies, such as Rarest First, Latest Useful Chunk First, and Greedy.

ACKNOWLEDGMENT

The reported study was partially supported by the Russian Foundation for Basic Research, research project No. 13-07-00953-a, 14-07-00090-a, and the Ministry of education and science of Russia, project 8.7962.2013.

 ISBN: 978-1-61208-318-6

REFERENCES

[1] E. Setton and B. Girod, "Peer-to-Peer Video Streaming. Springer," 2007, 150 p.

[2] Xuemin Shen, Heather Yu, John Buford, and Mursalin Akon, "Handbook of Peer-to-Peer Networking. Springer," 2010, 1421 p.

[3] Guangxue Yue, Nanqing Wei, Jiansheng Liu, Xiaofeng Xiong, and Linquan Xie, "Survey on Scheduling Technologies of P2P Media Streaming," Journal of Networks, Vol. 6, No. 8, August 2011, pp. 1129-1136.

[4] B. Fan, D. Andersen, M. Kaminsky, and K. Papagiannaki, "Balancing Throughput, Robustness, and In-Order Delivery in P2P VoD," In Proc. ACM CoNEXT, Dec. 2010.

[5] F. Clevenot and P. Nain, "A Simple fluid model for the analysis of the squirrel peer-to-peer caching system," Proc. of the IEEE INFOCOM, 2004, pp. 1-10.

[6] François Baccelli, Fabien Mathieu, and Ilkka Norros, "Performance of P2P networks with spatial interactions of peers. Networks and Telecommunications Networks, Systems and Services, Distributed Computing," Equipes-Projets GANG, TREC, Centre de recherche INRIA Paris, Rapport de recherche, n°7713, August 2011, pp. 1-23.

[7] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang, "A performance study of BitTorrent-like peer-to-peer Systems," Proc. of the IEEE Int. Conf. on Communication, Vol. 25, № 1, 2007, pp. 155-169.

[8] R. Srikant and D. Qiu, "Modeling and performance analysis of BitTorrent-like peer-to-peer networks," Proc. of the ACM SIGCOMM Computer Communication Review, 2004, pp. 367-378.

[9] L. Kleinrock and S. Tewari, "Analytical model for bittorrent-based live video streaming," Proc. of the IEEE CCNC, 2007, pp. 976-980.

[10] R. Kumar, Y. Liu, and K. W. Ross, "Stochastic fluid theory for P2P streaming systems," Proc. of the IEEE INFOCOM, 2007, pp. 919-927.

[11] Yipeng Zhou, M. Dah, Chiu, and J. C. S. Lui, "A Simple Model for Analyzing P2P Streaming Protocols," Proc. of IEEE Int. Conf. IN Network Protocols (ICNP 2007), Oct. 19, 2007, pp. 226-235.

[12] Y. Zhao and H. Shen, "A simple analysis on P2P streaming with peer playback lags," Proc. of the 3rd International Conference on Communication Software and Networks (IEEE ICCSN 2011), May 27-29, 2011. Xi'an, China, pp. 396-400.

[13] A. Adamu, Yu. Gaidamaka, and A. Samuylov, "Analytical modeling of P2PTV network," Proc. of the 2nd International Congress on Ultra Modern Telecommunications and Control Systems (IEEE ICUMT 2010), Oct. 18-20, 2010, Moscow, Russia, pp. 1115-1120.

[14] A. Adamu, Yu. Gaidamaka, and A. Samuylov, "Discrete Markov Chain Model for Analyzing Probability Measures of P2P Streaming Network," Lecture Notes in Computer Science. Germany, Heidelberg: Springer. 2011. Vol. 6869. pp. 428-439.

[15] Yuliya Gaidamaka and Andrey Samuylov, "Analytical Modeling of Playback Continuity in P2P Streaming Network with Latest First Download Strategy" // Lecture Notes in Computer Science, Germany, Heidelberg, Springer-Verlag, 2013, Vol. 8121, pp. 363-370.

[16] L. Ying, R Srikant, and S Shakkottai, "The Asymptotic Behavior of Minimum Buffer Size Requirements in Large P2P Streaming Networks", Information Theory and Applications Workshop (ITA), 2010, pp. 1-6.

 ISBN: 978-1-61208-318-6

Virtualization Model of a Large Logical Database for Diffused Data by Peer-to-Peer Cloud Computing Technology

Takeshi TSUCHIYA, Tadashi MIYOSAWA, Hiroo HIROSE
Faculty of Business Administration and Information
Tokyo University of Science, Suwa
Chino City, Nagano, Japan
Email: {tsuchiya.takeshi, miyosawa, hirose }@rs.tus.ac.jp

Keiichi KOYANAGI
Faculty of Science and Engineering
Waseda University
Kitakyushu City, Fukuoka, Japan
Email: keiichi.koyanagi@waseda.jp

***Abstract*—In this paper, we propose the use of peer-to-peer cloud computing technology to integrate databases that are distributed (diffused) throughout the Internet. We also propose the use of a large, virtualized, logical database that is managed using Structured Query Language (SQL). Combining databases into larger collections of data-"big-data" can make possible wonderful new services. Our proposed model has two main characteristics: First, SQL controls and manages data relationships using relational database management systems and key value stores, but discards location information. Second, service is made scalable by collaborations among distributed nodes. From the results of our evaluation, our model has sufficient service scalability for collaboration between distributed nodes, but not sufficient performance for big-data platforms.**

***Keywords*-Distributed Databases, Peer-to-Peer, Cloud Computing, Service Virtualization**

I. INTRODUCTION

Recent increases in data traffic from many type of web services indicate that the Internet services are being generalized and diversified. Almost all databases for web services use the relational database management system (RDBMS) model, which manages several types of contents concentrated on a specific nodes, but does nott allow for easy scaling of services. In particular, this model provides atomicity, consistency, isolation, and durability (ACID) characteristics for data and services. However, the scalability of node distribution is limited to two or three nodes in master/slave relationships. The master node manages all data exclusively. Therefore, the master node cannot usually process amounts of data, such as more than 10,000 request per second or operations more than a million data. However, this model includes some useful functions based on ACID characteristics for processing large amounts of data.

In recent years, several types of databases have come into use, which use the "No SQL" model. One type is a key value store (KVS) database, which focuses on service scalability and continuity [2][3]. No SQL databases provide service scalability, load balancing, and high availability that can scale out manner. However, they do not have some basic but useful functions that are provided by RDBMS databases, such as transactions, complicated retrieval by conditions, and aggregate processing. These new database services continuously involve the above functions in the view of service and require to construct same functions as middleware of application. These database services distributed throughout the Internet and are distinguished into the mentioned two types. These perform the same service functions as data management services, but have a different attitude. Therefore, it is difficult to unify all databases on the Internet according to either model. There can also be several types of models of the Internet services.

Currently, several services such as social networking services (SNSs) correct and manage No SQL databases with large logical spaces. The databases are used for recommendation services and Web-marketing information. The types and quality of information that can be retrieved from each service are limited and depend on each service. Sharing of contents data among services have possible to generate the new points of view from these. However, generating new points of view requires processing and analyzing data that have been combined from many databases in each service, which is difficult for No SQL databases to do adaptively for the above-mentioned reason. During processing, complex data must be moved and merged into a new No SQL database for control and analysis.

This paper proposes the use of peer-to-peer (P2P) cloud-computing technologies to create a logical data management space for integrating databases that are distributed throughout the Internet. Every user can acquire and control all data connected to the proposed logical space and generate schema information and transactions among the distributed databases. In other words, the proposed model virtualizes these database and shares integration data management in same way. Therefore, following the process out lined in this this paper should improve the quality of database service by integrating distributed data throughout the Internet, scaling up RDBMS functions, and clarifying the use of big data.

This paper is devided into five parts; Section 2 discusses the proposed model based on requirements from the analysis of current databases. The results are evaluated in Section 3. Section 4 discusses possibility and usage as system. Section

 ISBN: 978-1-61208-318-6

5 concludes this paper.

II. Logical Integration Model for Disributed Databases

This section discusses and propose our overlay network model for distributed databases over the Internet.

A. Requirements for Model

Several types of databases are used for network services, which differ in functions and characteristics. Our proposal method enables access to all data on these databases without minds of their belonging and allows them to be managed independently. Nodes must be flexible, adapting to service conditions such as the No SQL model, for big-data service, and control functions, such as RDBMS model. The interface protocol of the proposed model also affects usability for developers and current resources. Therefore, its interface is is expected to use similar the current SQL. Each database service on the Internet is independently are managed under each definition of policy for security in advance. The method of management in the proposed model must overcome and manage these differences. Therefore, our model can flexibly adapt policies from the logical space.

B. Creating Logical Spaces Via Node Collaboration

This section discusses and clarifies the method of constructing a logical space using distributed computing technology.

The logical overlay networks among databases are not provided by static servers but are constructed by collaboration among nodes in regards to service scalability. We have discussed about P2P distributed platform technology in [1] and [8]. All nodes and objects on this overlay network are identified by 128-bit IDs. This platform provides flexible and dynamic node management for general normal nodes to demand for features such as node movement with continuous services and scaling up using P2P cloud computing technologies. This platform also manages data or pointer information for data on the databases of component nodes.

Nodes ware selected for the overlay network based on the author's presidential research in [8]. Under this algorithm, nodes are sequentially selected on the basis of their conditions and performance: they are divided into some roll on overlay network demand of service situation. This election algorithm is defined as a combination of complex calculations based on the features of services such as stability of service, network quality, and processing power. These selected nodes are forming the best-effort overlay network.

C. Method of Managing Distributed Data

This section discusses the mapping of distributed data on the overlay network and clarifies how to manage data on distributed nodes.

All distributed data and tables on databases over the Internet are mapped to the overlay network in the following way, and they are virtualized as a single large database. Selected component nodes behave as service managers of the logical overlay network and adjust the range of the overlay network. The area of the network is 64bit by 64 bit –two-dimensional logical space. Mapped data and tables are allocated in this range, and the range is identified by a unique 128bit object ID. Each piece of data can be easily accessed using this range ID and sequential coordinate information. This ID is generated in the form "URI + the original name of the database (table). " The hashed value of the key is the object ID, and it points to an area on the overlay network. This point, called the starting point, is shown in Fig. 1 at the top of mapping, and mapped data and tables are allocated after it. Therefore, all starting points can be acquired above mentioned two types of information on the large overlay network. Applications and users are easy to derive. The range of mapped data is adjusted and allocated by demand of their quantity. This method is described in section II-D.

Each starting point manages the mapping range of data and information from the original database. The information from all starting points is shared as cashed information among component nodes of the overlay network. The range of mapped data can over lap in another range on the overlay network, such as the starting point, and mapped data using the ensured range for each service. When the expectant range for mapping data has already been ensured by other mapped data or tables, the derivation of the expected starting point uses "object ID + table name + 1" as the key of a hashed function. As a result, all starting points are displaced the range ensured others. New expected range need to place without ensuring as other range. The derivation of the starting point is incremented iteratively: "object ID + table name + 2" until the range for data mapping is ensured.

The size of the mapping range is not confined to applications and users. Allocated size for mapping is assumed to determine the amount of mapped data on current databases and estimated additional data in services. When data is unexpectedly incremented, the new range is added to the current range, and the platform updates range information managed at the starting point. Although frequent addition to the range makes data distribution unbalanced, the proposed overlay network is adapted to the function for the re-allocation of area or for autonomous scaling bases on demand and situation.

Data and tables having in data relationships such as an RDBMS are mapped to ensured ranges with their data relations. Data without RDBMS relationships are mapped to there by sequence of each object ID.

D. Security Method for Data Mapping

Updating between mapped data and original data in each distributed database is restricted in the following two ways to ensure data consistency.

 ISBN: 978-1-61208-318-6

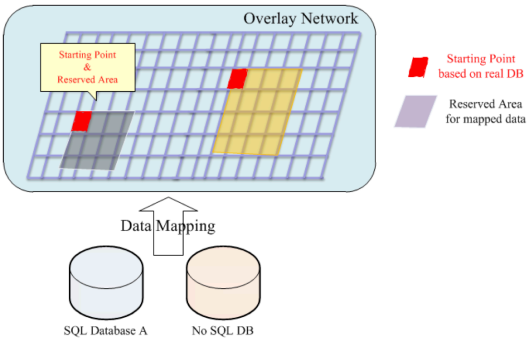


Figure 1. Data Mapping on an Overlay Network

(i) Data management is allowed only for mapped data by overlay network functions with interdicting original data on distributed databases.
(ii) The direction of data control flow is limited: only original data can be update and added, and only overlay network function can adjust the read and controls for mapped data.

Although data management in our proposed model has these restrictions, it satisfies the eventual consistency model discussed in P2P distributed services, and ensures data consistency in some range.

The restriction on data management, restriction (i) assumes that all management of mapped data is shared among applications and users such as in current RDBMS service. This restriction is in regards to the use of the overlay network as a large database.

For restriction (ii), an instance of the process monitoring change of the original data is allocated on the neighbor of the original databases. Although this manner sensitively monitors the differences, the delay of data consistency is less than with the previous method. This method also allows the adaptation of current important service data called big data to the overlay network easily. In particular, the distribution of current data among No SQL model databases enables the generation and management of relationships among data.

E. Management of Coordinate Spaces as One-Dimensional Information

Mapped data on an overlay network is managed on a 64bit×64bit range of a two dimensional logical coordinate space which is similar to an RDBMS model database. Managing this two-dimensional space to the distributed algorithm is necessary for management by collaborations among selected nodes. It uses Z-Ordering [9] for reducing dimensional information. This manner makes 64bit ×64bit two-dimensional logical information converting to 128 bit one-dimensional information. This 128 bit value corresponds to the point of two-dimensional coordinates. All nodes with the same platform can derive this reduced value based on

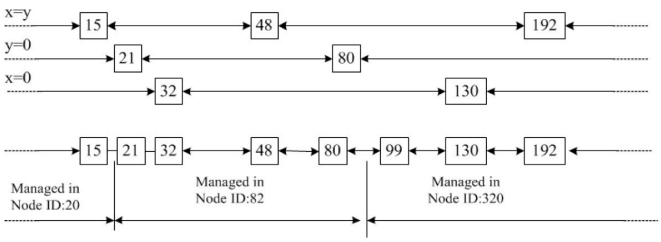


Figure 2. Management of Reduced Coordinate Information

that cordial information for the allocation of data without inquiry to other nodes. The use of this algorithm reduces load without requiring derivation protocol.

The reduced coordinate information is managed by the structure of a layered list such as a Skip List [10] among nodes shown in Fig. 2. Each component node is described sequentially by node ID in Fig. 2. They only manages node ID information of predecessor and successor node.

The three layers from the top layer shown in Fig. 2 provide the list information of the horizontal axis, vertical axis, and diagonal axis on based on two-dimensional information. Managing the information from typical coordinate points affects the efficiency of the acquisition of data among distributed nodes that manage the above mentioned list information.

The bottom layer in Fig. 2 is the list structure of all coordinate points. When the data is allocated to this coordinate points, this list adds information about mapped data or pointer information to the data. Component nodes divide the list information of bottom layer in the order of their node ID. Each node only knows and manages some objects with smaller object IDs than themselves as same as ring management manner of Chord [11]. Component nodes enable the inference of who manages objects sweepingly by object ID without inquiry into other nodes. This protocol reduces the ability to acquire coordinate information efficiently. At the same time, the distribution of an object ID would not concentrate in a typical range of space using a hash function

F. Interface for Applications

The SQL commands generated from applications are received and analyzed at the interface layer shown in Fig. 3. The data management method from users and applications discussed in section II-C enables the attainment of mapped data using start point information which hashed object ID and table name. This overlay network corrects information from mapped databases using received commands among distributed databases and enables the generation of new tables mapped to themselves. Therefore, our proposed model enables the provision of data management service such as RDBMS without the distribution of databases.

 ISBN: 978-1-61208-318-6

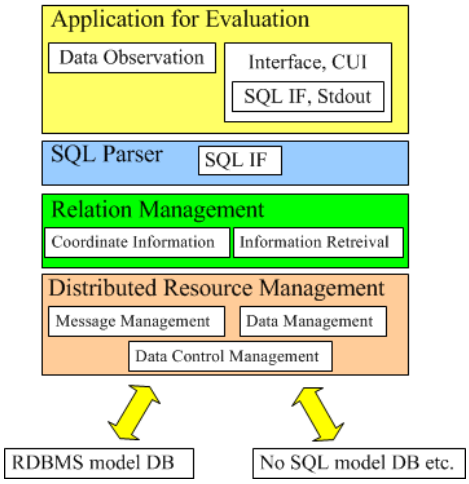


Figure 3. Current Implementation for Evaluations

Table I
IMPLEMENTATION ENVIRONMENT

OS	Windows 7 Professional
CPU	Xeon 2.4GHz×2
RAM	4 GB
DB	My SQL 5.1.44
DevLang	Java SDK 1.6

G. Implementation

The above mentioned fundamental functions are implemented as shown in Fig. 3. This model works as middleware and provides the mentioned data management functions for several types of applications. This implementation model is depicted in a Table I environment and evaluated below. The logical overlay network is composed of three layers, and the outline of the functions of each layer is described in Fig. 3. In this evaluation, the application is set for a specialized e-commerce service.

III. EVALUATION

The proposed logical database model is evaluated on the basis of the serviceability and availability of e-commerce that is implemented using the model.

A. Emulation Scenario

Evaluation using real nodes and environment is difficult because of their distribution scalability, and a simulation can indicate the abilities of the proposed model in an applicable distributed environment. We conduct simulations of implemented nodes.

The simulation environment is constructed by two nodes, each of which manages a hundred other nodes. This environment enables us to dynamically construct the network at the application layer such as cloud computing with hundreds of order of nodes. Generally speaking, the common RDBMS model can be composed of a few distributed nodes at most. On the other hand, the No SQL model is composed of lots of distributed nodes without a limitation – what is called a ”scale free” model. Therefore, this simulation environment is necessary because of the hundreds of nodes. Our evaluation in this paper is constructed by two hundred distributed nodes.

Nodes are executed as independent instances on the Internet, and they behave independently based on their data. All interactions in this environment use the same protocols and methods as the Internet protocol, and all instances are allocated to a server by a unique node ID: Instances with odd node IDs are distributed to one node, and instances with even node IDs to the other. Therefore, communications among the nodes (instances) configured in the skip list shown in Fig. 2 generate communication traffic between these servers and enable the simulation of realistic situations. Data management on these nodes is connected to six independent databases of the RDBMS model,as shown in Fig. 3. All the including data and tables are mapped to the proposed logical overlay network and then evaluated.

B. Simulation Environment

This simulation uses the TPC-W [12] standard for e-commerce services, which was created by the Transaction Processing Performance Council (TPC).

Our proposed model integrates six independent databases for e-commerce services using cloud technologies and provides access to all data on these databases without caring about location of its belongings. All of these integrated databases use the RDBMS model, and they include management information for each service entity, such as a customer or commodity.

Simulating user accesses to integrated overlay network in the same time, more than a hundred processes are executed on this simulation environment. In this case, a process corresponds to a user. Each user (process) starts to access web page starting with the top one and retrieves some types of commodities using queries randomly generated on the overlay networks. Each page that is retrieved results includes one-hundred commodities per page. If the results of retrieval are a thousand commodities, 10 pages are generated. This evaluation clarifies the elapsed time from the generation of the process to the generation of the pages, including the results of the retrieval query as a response. This evaluation also discusses the serviceability and availability of the proposed overlay network approach for distributed databases.

C. Evaluation Results

The following sections are discussed and include the results of evaluations for technology points.

1) Availability and Responsiveness of database service: Regarding the serviceability of node distribution as a database service, Fig. 4 compares two types of average

 ISBN: 978-1-61208-318-6

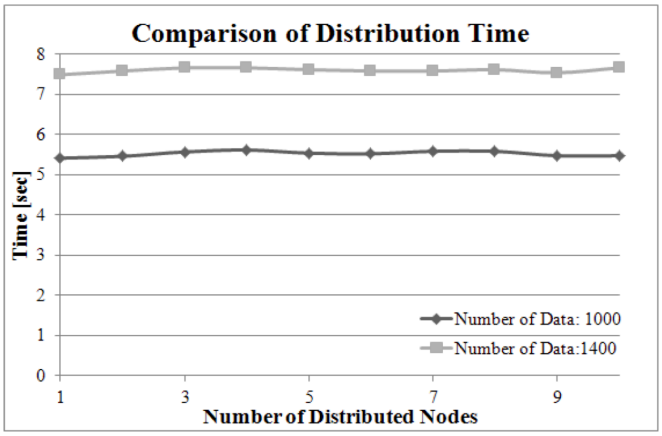


Figure 4. Relation of data quantity and node distributions

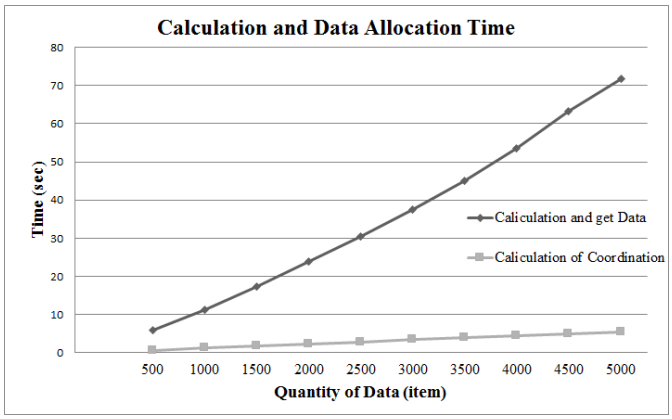


Figure 5. Serviceability of proposed model

elapsed time, which means the difference between pages generated in response to a query (10 and 14 pages) including results (1,000 and 1,400). The x axis shows the number of distribution nodes, and the y axis shows the time until generating response pages from the first access. These curves in the graph indicate similar trends and stable lines until 10 nodes unrelated to node distributions. These trends have continued over 200 of node distributions, although the lines are not described in this graph. The graph shows that node distribution does not affect the serviceability of the proposed model caused by the control and management of data. The graph also shows that the elapsed time is relative to the increase in the number of results (data items) from $1,000$ to $1,400$ data. The graph indicates that time can be incremented by about 85% in both cases caused by node distributions. From the result, service scalability caused by node distribution data does not affect the protocol and serviceability in the logical area.

2) Serviceability:: The data management method for overlaying the network discussed in sec. II-C used two types of method due to limitation of data replication. Fig.5 indicates the comparison of these manners and each characteristic. The x axis shows the number of retrieved results, and the y axis shows the elapsed time for processing. In Fig. 5, the line (a) indicates the case for managing the mapped data on an overlay network, and the line (b) indicates in the case of managing pointer information of data on it without themselves.

Fig. 5 shows that the elapsed time for processing is increased by the increment of the target data in both cases. In specific, the increment rate of line (b) is significantly larger that of line (a). This difference between these lines is caused by the amount of processing elapsed time from sending the query to generating the results. In the case of line (b), the query sent to the overlay network for retrieval is independently distributed to each databases, and each database manages the data entity mapped to pointer information. In comparison with line (a), the divide between a query to a database and the collection of results from a database takes elapsed time for generating retrieval results.

IV. Considerations

This section discusses the characteristic and usage situations of the proposed model as a as a realistic system based on the evaluation results and model design.

A. Database Service

What follows is some discussion of functionally and serviceability.

Relationship management among pieces of data: Users map and manage data without reference to where data is located. Data is retrieved and generated across overlay networks. In other words, the proposed model allows relationships to be generated for pieces of data on distributed databases and then allows applications to use those relationships without moving data to a single database. In particular, some implementations of RDBMS database manage big data using a No SQL model. Service Data is often divided into some shards to retain read-write capability and improve response times. For data managed using No SQL, relationships between pieces of data and tables cannot be generated and managed unless the change of type of database. However, mapping all data to an overlay network enables the relationships to be generated and managed without unifying these distributed databases. Thus, our proposed model provides the integration of big data services effectively without database re-construction.

Availability as a database service: The proposed overlay network is composed of distributed databases that use a P2P platform. The amount of data and the number of component databases can be changed flexibility and dynamically: data and databases can be added, changed and removed. The evaluation results show that the quality of the proposed

 ISBN: 978-1-61208-318-6

model does not affect the number of distributed node. Thus, the overlay network provides a continuous and stable service by demand of its situations autonomous adjustment among component nodes for some event: load of node and outage from the overlay network caused by the increment of traffic on the overlay network. This method provides more availability than the current RDBMS model, making it more flexible and autonomous. Similar with No SQL, the proposed model adopts a P2P distributed model and it can be expected to provide similar service availability.

Usage in big-data situations : The proposed model enables the generation and management of relations among pieces of data without an ordinal model and allocations for big-data generated from SNSs and web services. Developer do not need to divide into new shard and they apply conventional relations of data to new services when service providers expand or improve service. Therefore, the proposed model is effective and useful for current big-data services.

B. Future Work

The above section clarified the characteristics of our proposed database service in comparison with the current RDBMS model in regards to serviceability, scalability, and availability. The model also provides important function for services. However, evaluation results indicate that the delay (lag) for transactions and response to service are urgent issues. The improvements provided by the implementation are limited by the distributed system. Therefore, future work will apply a management algorithm to the overlay network spaces for the replication or caching of distributed well-accessed data described in section IV-A to improve the implementation. Our proposed model should be feasible for use on real services.

V. Conclusion

We proposed and discussed a means for integrating databases that are diffused (distributed) throughout the Internet. The databases are integrated using peer-to-peer cloud computing technology to construct a large logical database space. All data in the distributed databases is integrally controlled by SQL regardless of the types of component databases (such as RDBMS, or KVS) and the size of the databases. These improvements make possible integrated platform technology for web services and new applications such as big-data services. However, the current implementation cannot be immediately applied to such services due to low performance during situations. The simulation suggests that the model can be used for big-data services and can be improved by adding function such as data caching, and replications. The model must be improved before next steps can be taken .

Acknowledgment

This research was partially supported by Strategic Information and Communication R&D Promotion Program (SCOPE) of the Ministry of Internal Affairs and Communications, Japan, Grant number:122304003 .

References

[1] T. Tsuchiya, H. Sawano, M. Lihan, H. Yoshinaga, and K. Koyanagi, “A Distributed Information Retrieval Manner Based on the Statistic Information for Ubiquitous Services”, Progress in Informatics No. 6, pp. 63–78, April 2009

[2] F. Chang et al., “Bigtable: A Distributed Storage System for Structured Data”, ACM Transactions on Computer Systems, Vol. 26 Issue 2, June 2008

[3] W. Vogels, “Eventually consistent”, Communications of the ACM Rural engineering development, Vol. 52 Issue 1, January 2009

[4] A. Lakshman, and P. Malik , “Cassandra: a Decentralized Structured Storage System”, ACM SIGOPS Operating Systems Review, Vol. 44 Issue 2, April 2010

[5] ”MongoDB”, 31 January 2014, <http://www.mongodb.org/>

[6] E. Meijer, and G. Bierman, “A co-Relational Model of Data for Large Shared Data Banks”, ACM Queue, Vol.9, Issue 3, pp. 1–19

[7] U. F. Minhas et al., “Elastic Scale-out for Partition-Based Database Systems”, Proc. of Int. Self-managing Database Systems, Washington, DC, April 2012

[8] H. Yoshinaga, T. Tsuchiya, and K. Koyanagi, “Coordinator Election Using the Object Model in P2P Networks”, 3rd International Workshop on Agents and Peer-to-Peer Computing, Springer Berlin/Heidelberg, Vol. 3601, pp. 161–172, New York, 2004

[9] G. M. Morton, “A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing”, Technical Report, IBM Ltd., Ottawa, Canada, 1966

[10] W. Pugh, “Skip lists: a Probabilistic Alternative to Balanced trees”, Communications of ACM 33, pp. 668–676, June 1990

[11] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, ACM SIGCOMM, pp. 149–160, San Diego, CA, 2001

[12] TPC-W, ”TPC Transaction Processing Performance Council”, 31 January 2014, <http://www.tpc.org/tpcw/>

 ISBN: 978-1-61208-318-6

Adaptive Online Compressing Schemes Using Flow Information on Advanced Relay Nodes

Mei Yoshino, Hiroyuki Koga
Graduate School of
Environmental Engineering
University of Kitakyushu, Japan
Email: moeu@net.is.env.kitakyu-u.ac.jp
h.koga@kitakyu-u.ac.jp

Masayoshi Shimamura
Global Scientific Information
and Computing Center
Tokyo Institute of Technology, Japan
Email: shimamura@netsys.ce.titech.ac.jp

Takeshi Ikenaga
Graduate School of Engineering
Kyushu Institute of Technology, Japan
Email: ike@ecs.kyutech.ac.jp

***Abstract*—As the number of users and applications continues to grow, Internet traffic is growing explosively. Excessive traffic causes network congestion, though, and significantly degrades communication performance. In this paper, we propose adaptive online compressing schemes that use flow information on advanced relay nodes to efficiently reduce the amount of traffic by utilizing network and computational resources. The proposed schemes compress multiple packets forwarded in the same direction by utilizing the waiting time. Furthermore, we evaluate the proposed schemes compared to an adaptive packet compression scheme previously proposed.**

Keywords-adaptive online compression; advanced relay node; network resource; computational resource

I. INTRODUCTION

Continual growth in the number of users and the frequent data exchange of content such as videos and music are causing Internet traffic to increase explosively. According to [1], mobile data traffic will grow at a compound annual growth rate of 66 percent from 2012 to 2017, reaching 11.2 exabytes per month by 2017. When traffic becomes excessive it causes network congestion, which in turn significantly degrades communication performance. Since network resources are limited, they must be used efficiently to alleviate this problem.

To enable efficient use of network resources, an adaptive packet compression scheme has been proposed [2]. This scheme assumes that advanced relay nodes are located inside networks and that these nodes possess not only network resources (i.e., forwarding functions) but also computational resources (i.e., processing functions) [3]. This scheme compresses an incoming packet at the advanced relay nodes while the packet is waiting in an output queue to transfer when congestion occurs. The authors showed numerical results in terms of compression ratio using a data set from an actual network and confirmed the effectiveness of the adaptive packet compression scheme [2], [3]. Even though the adaptive packet compression scheme could reduce the data size by only 5% (i.e., a compression ratio of 0.95), it improved the packet discard ratio and delay time.

In this paper, we suggest adaptive online compressing schemes that use flow information on advanced relay nodes to improve communication performance by reducing the compressed data size more effectively. A key idea is that the proposed schemes compress a block generated from multiple packets forwarded in the same direction (e.g., towards the same destination host or the same subnet) in an output queue at advanced relay nodes. For the block compression, we used a previously reported approach. In [4], to shrink the data size of archival traffic dump data, the authors focus on correlations between header fields among multiple packets. They then show that the compression ratio can be improved by rearranging header fields so as to store similar fields into a single block. In our case, since we compress a block of multiple packets going in the same direction, these packets have similar header fields (e.g., the destination IP address). Therefore, we expect the compression ratio to be improved. However, if the proposed schemes attempt to compress a large block generated from many packets, the compression opportunity can be lost. This is because the proposed schemes cannot gather the packets before the packets are transmitted from an output queue. To efficiently compress a block, we propose two compression schemes: (1) a flow compression scheme which compresses packets having the same 5-tuple header information, and (2) an edge compression scheme which compresses packets passing through the same egress edge of advanced relay nodes. Furthermore, we investigate the effect of the number of compression packets and the compression time when the proposed schemes are used. Through simulations, we show the potential and effectiveness of the proposed schemes.

The remainder of this paper is organized as follows. In Section 2, we describe related studies in terms of data compression. In Section 3, we explain the proposed schemes. We describe the simulation environment in Section 4 and the simulation results in Section 5. We conclude in Section 6.

II. RELATED WORK

As stated in Section 1, several data compression schemes have been proposed. In this section, we first describe the adaptive packet compression scheme, which is the basis of our proposed schemes. We then describe IPzip from a perspective of efficient multiple packet compression.

 ISBN: 978-1-61208-318-6

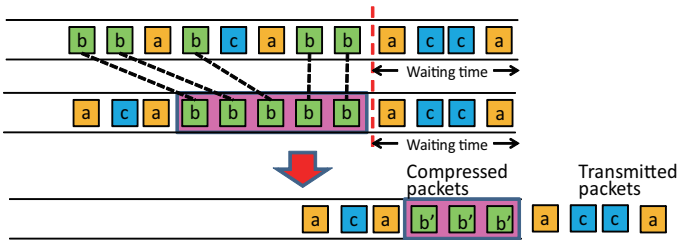


Figure 1. Behavior of the proposed schemes

A. Adaptive packet compression scheme

The adaptive packet compression scheme [2] aims to improve communication performance by efficiently using both network and computational resources. When an advanced relay node receives a packet, the node calculates the waiting time of the packet inside its output queue, and then it decides to compress the packets if the waiting time is sufficiently large. Since it compresses packets by exploiting the waiting time, the processing time of compression becomes nearly zero. Therefore, the adaptive packet compression scheme achieves better online packet compression at advanced relay nodes.

Through data analysis using an actual data set, the authors found that packets can be classified into compressible and incompressible packets. They showed that the average compression ratio of all the packets was 0.945 and that of the compressible packets was 0.929. These results showed that actual traffic volume could be reduced by packet compression even if the compression ratio was high (i.e., less than a 10% reduction of data size).

In [2], the authors also showed the effectiveness of the adaptive packet compression scheme through simulation evaluation. In this evaluation, the compression ratio was set to 0.95. Simulation results showed that the adaptive packet compression scheme improved the packet discard ratio and delay time even though it could reduce the data size by only 5%.

B. IPzip

IPzip [4] compresses a block created from multiple packets and is used to reduce the data size of stored traffic dump data. The authors focus on similarities among these packets. For example, if packets are transferred to the same destination, these packets have the same destination IP address in their header fields. IPzip rearranges header fields inside stored dump data so as to collect the same or similar information inside header fields, and then it compresses all the data that has rearranged header fields. IPzip can achieve better compression with a low compression ratio through this sophisticated compression approach.

III. Adaptive Online Compressing Schemes

In this section, we describe adaptive online compressing schemes. We first describe an overview of our idea and then propose two kinds of compression scheme.

A. Overview

Unlike the adaptive packet compression scheme [2], our proposed schemes gather multiple packets adaptively in an output queue at advanced relay nodes and compress them by utilizing the waiting time. Various criteria can be used to create blocks: (1) flow (i.e., same source and destination IP addresses, source and destination port numbers, and protocol number), (2) service (i.e., same destination address and port number, and protocol number), (3) host-by-host (i.e., same source and destination addresses), and (4) destination group (i.e., same network address). The proposed schemes compress multiple packets forwarded in the same direction using flow and destination group information (i.e., 5-tuple header information or information of passing through the same egress edge of advanced relay nodes). Fig. 1 illustrates an example of the proposed schemes' behavior. In this figure, multiple packets forwarded in the same direction (packets b) are grouped and compressed while they are waiting in the output queue. To compress a block generated from multiple packets, our proposed schemes need more processing time than is needed for a packet compression scheme. We define the time needed to compress a block as the "compression time". Moreover, we define the number of packets to be compressed as the "number of compression packets". Let n be the compression time and m be the number of compression packets. The proposed schemes compress m packets forwarded in the same direction when the queue length is more than $n+m$ packets. Note that we normalize the compression time using the packet transmission time, so that we represent the number of packets as the compression time.

B. Compression schemes

To efficiently compress a block, we propose two types of block compression: a flow compression scheme and an edge compression scheme. Both generate a block from multiple packets having the same information in a part of the header field. However, the two schemes generate a block differently.

Flow compression scheme

The flow compression scheme generates a block by using information related to end nodes. This scheme gathers multiple packets having the same 5-tuple header information (i.e., source and destination IP addresses, source and destination port numbers, and protocol number) in the output queue, and generates a block from these.

Edge compression scheme

The edge compression scheme gathers multiple packets passing through the same egress edge of an advanced relay node in the output queue, and generates a block from these. Therefore, this scheme compresses a block containing different transport flows.

IV. Simulation Environment

In this section, we evaluate the proposed schemes in comparison with the adaptive packet compression scheme through simulations. First, we describe the simulation model and evaluation indices. We use network simulator ver. 2.35 [5] after implementing the proposed schemes.

A. Simulation model

Fig. 2 shows the network topology. In this simulation model, congestion can occur at links between an ingress edge

 ISBN: 978-1-61208-318-6

TABLE I. SIMULATION PARAMETERS

Buffer size on each node	50 [packet]
Transport layer protocol	TCP with SACK option
Packet size	1500 [Byte]
Number of compression packets	1–20
Compression time	5–30 [packet]

of advanced relay nodes and a core router. A TCP sender node $S_{i,j}$ sends packets toward a TCP receiver node $R_{i,j}$ connected to an egress edge node E_i, where i represents the number of ingress and egress edge nodes and j represents the number of end nodes connected to a single edge node. The links between each ingress or egress edge node and core routers have a bandwidth of 100 Mb/s and a delay time of 3 ms, while the bottleneck link between core routers has a bandwidth of 200 Mb/s and a delay time of 5 ms. All other access links between each sender or receiver node and the ingress edge or egress edge nodes have a bandwidth of 100 Mb/s and a delay time of 1 ms. The proposed schemes compress packets at ingress edge nodes and expand them at egress edge nodes. If the compressed packets are lost, ingress edge nodes retransmit them.

As the simulation parameters, we set the compression ratio of the adaptive packet compression scheme to 0.95 in accordance with [2]. To determine the compression ratio of the proposed schemes, we preliminarily investigated the compression ratio of multiple packets (from 1 to 100) using the Lempel-Ziv-Oberhumer (LZO) compression algorithm with a data set from an actual network (4.7 GB, approximately 50 million packets). Through this investigation, we found that the compression ratio varied approximately from 0.25 to 0.95. In this simulation, using the mean values of the preliminary results, we set the compression ratio of the proposed schemes to 0.6 or 0.5 when the number of compression packets is 5 or 10, respectively. Other simulation parameters are summarized in Table I.

We investigate the effect of the number of compression packets on communication performance. In this scenario, the number of end nodes pairs varies from 9 to 300 (multiplies of three) and a single TCP flow will flow between each pair of end nodes, so there are 9 to 300 TCP flows.

B. Evaluation indices

To evaluate the effectiveness of the proposed schemes, we focus on the total throughput performance as an evaluation index. The total throughput is calculated by summing the throughput of all TCP flows from 10 to 30 seconds after the simulation starts to avoid the influence of a transient period and it is averaged over 10 simulation runs with different random seeds. To analyze the results, we also investigate the number of compression processings.

V. SIMULATION RESULTS

In this section, we show evaluation results of the proposed schemes compared with the performance of the adaptive packet compression scheme. First, we investigate the throughput performance of each scheme. We then examine the effect of each parameter on throughput performance.

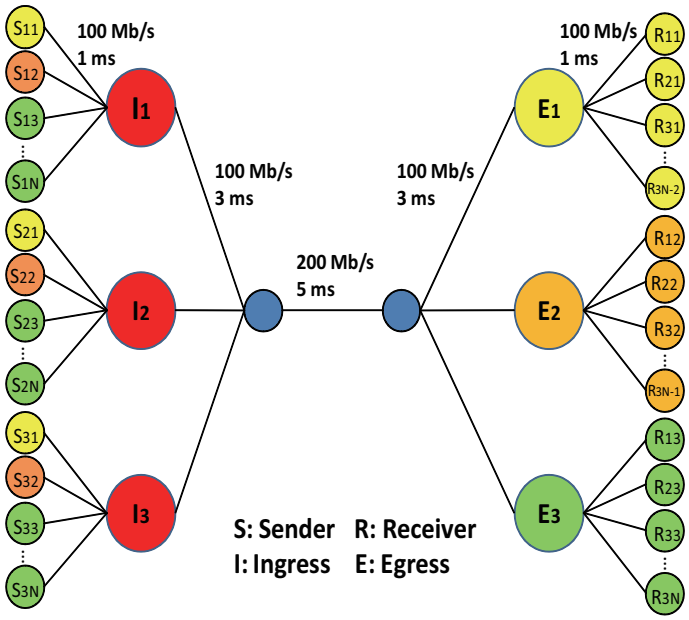


Figure 2. Simulation topology

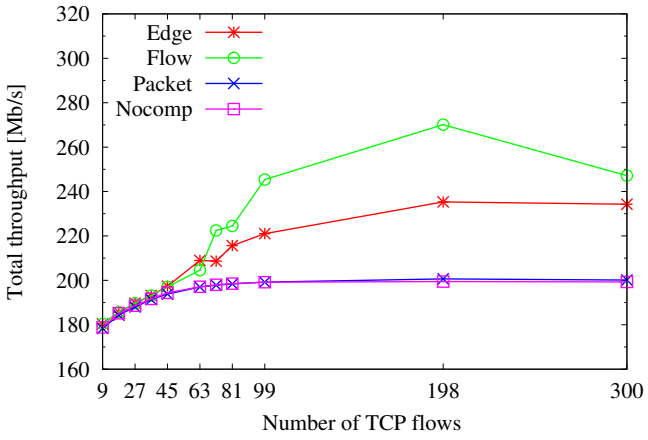
A. Throughput characteristics of each scheme

We first evaluate the throughput performance of each scheme. Fig. 3 shows the total throughput of the proposed schemes, the adaptive packet compression scheme, and a no-compression scheme (simply relaying all packets without packet compression) when the number of TCP flows varies from 9 to 300. In this figure, "Edge" denotes the edge compression scheme, "Flow" denotes the flow compression scheme, "Packet" denotes the adaptive packet compression scheme, and "Nocomp" denotes the no-compression scheme. The number of compression packets for the proposed schemes is set to 5 (the compression ratio is 0.6) or 10 (the compression ratio is 0.5), while the compression ratio of the adaptive packet compression scheme is set to 0.95, as described in the previous section. The compression time is set to the time needed to forward 5 packets.

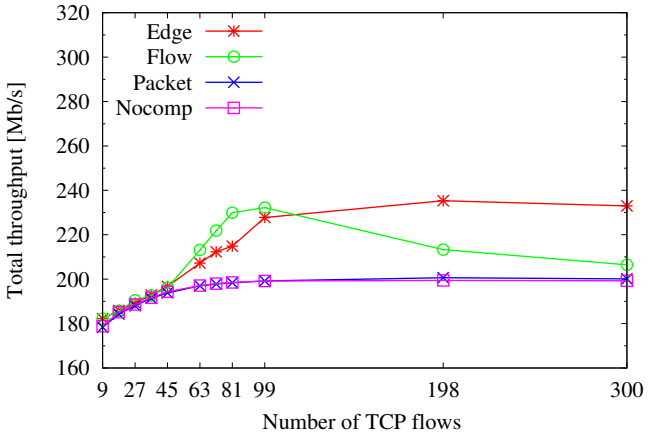
Figs. 3(a) and 3(b) show that the total throughput of the edge and flow compression schemes is higher than that of the other schemes regardless of the number of compression packets. The throughput of the proposed schemes exceeds the bottleneck link bandwidth by effectively compressing multiple packets, while that of the adaptive packet compression and no-compression schemes is limited by the bandwidth. In the case of a small number of compression packets, the flow compression scheme attains higher throughput than the edge compression scheme over a wide range of the number of TCP flows. However, the throughput of the flow compression scheme decreases as the number of TCP flows increases when the number of compression packets is large.

Let's investigate the reason for the above phenomenon. Figs. 4(a) and 4(b) respectively show the number of compression processings for each scheme when the number of TCP flows varies from 9 to 300 and when the number of compression packets of the proposed schemes is set to 5 or 10. The number of compression processings is approximately the same for the flow and edge compression schemes when the number

 ISBN: 978-1-61208-318-6

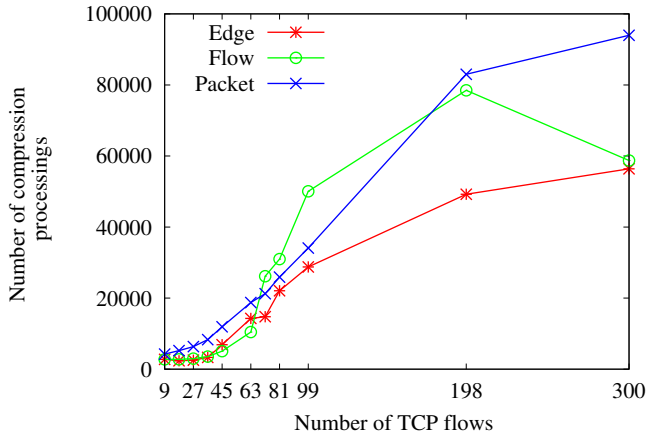


(a) Number of compression packets: 5

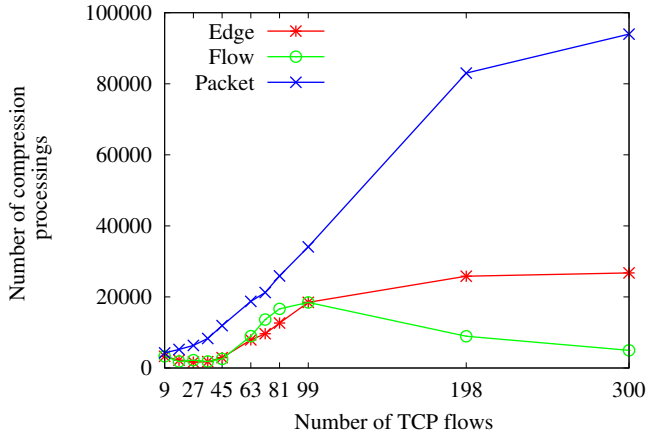


(b) Number of compression packets: 10

Figure 3. Throughput performance



(a) Number of compression packets: 5



(b) Number of compression packets: 10

Figure 4. Number of compression processings

of TCP flows is less than 63. On the other hand, the number of compression processings of the flow compression scheme falls as the number of TCP flows increases, especially in the case of a large number of compression packets, while that of the edge compression scheme increases as the number of TCP flows increases. This is because the flow compression scheme has difficulty gathering multiple packets having the same flow information due to the limited buffer size. Compared to the flow compression scheme, the edge compression scheme can compress packets much more often because each of the ingress edge nodes needs to handle only three types of packet going toward the egress edge nodes. Therefore, the edge compression scheme can better maintain throughput performance than can the flow compression scheme in the case of a large number of compression packets.

The above results demonstrate that the proposed schemes can improve throughput performance compared with the other schemes. In the following subsection, we discuss the effect of each parameter of the proposed schemes on the throughput performance.

B. Effect of each parameter

To analyze the performance between the flow and edge compression schemes in detail, we investigate the effect of each parameter on the throughput performance. The performance of the proposed schemes depends on how many packets are successfully compressed. Namely, the main factors determining performance are the number of compression packets and the compression time. In this subsection, we examine the effect of these parameters on the throughput performance and the number of compression processings.

First, we focus on the effect of the number of compression packets on the throughput performance. Figs. 5(a) and 5(b) show the total throughput and the number of compression processings of each scheme when the number of compression packets varies from 1 to 20, respectively. The number of TCP flows is set to 198, where the number of compression packets has a significant impact on the throughput performance of the flow and edge compression schemes as shown in Fig. 3. The compression ratio is set to the average value of 0.6 regardless of the number of compression packets in order to focus on the opportunity of compression in the flow and edge compression schemes. The compression time is set to 5 packets.

As shown in Fig. 5(a), the flow compression scheme enables excellent throughput when the number of compression packets is small, especially in the case of 5. However, the throughput of the flow compression scheme drastically decreases as the number of compression packets increases. On

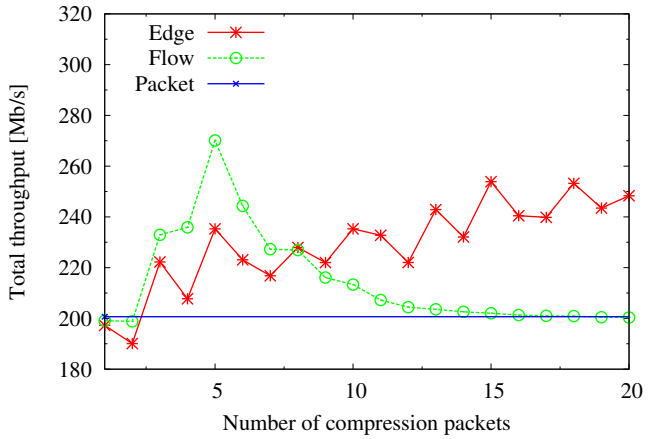
 ISBN: 978-1-61208-318-6

the other hand, the throughput of the edge compression scheme increases as the number of compression packets increases. Consequently, the edge compression scheme maintains high throughput over a wide range of the number of compression packets.

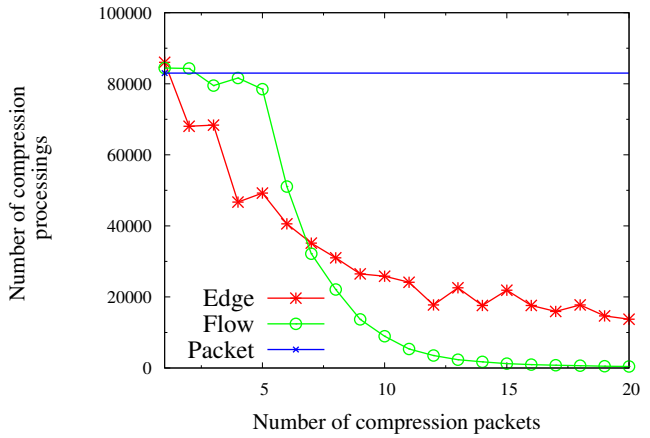
In order to understand the reason for this, let's consider the number of compression processings shown in Fig. 5(b). The number of compression processings in the flow compression scheme drastically decreases as the number of compression packets increases. This is because the flow compression scheme has difficulty gathering multiple packets having the same flow information due to the limited buffer size as discussed in the previous subsection. On the other hand, although the number of compression processings in the edge compression scheme decreases as the number of compression packets increases, it remains higher than that in the flow compression scheme when the number of compression packets is large. That is, compared to the flow compression scheme, the edge compression scheme can compress packets much more often because each of the ingress edge nodes has to handle only three types of packet going toward the egress edge nodes.

These results demonstrate that the flow compression scheme enables higher throughput than the edge compression scheme with a small number of compression packets, while the edge compression scheme maintains high throughput with a large number of compression packets. The flow compression scheme gathers multiple packets having the same 5-tuple header information; i.e., that belong to a flow. In contrast, the edge compression scheme gathers multiple packets passing through the same egress edge nodes; i.e., that belong to multiple flows. With a small number of compression packets, since the opportunity of compression between the flow and edge compression schemes is almost the same, the flow compression scheme can rapidly increase the throughput of the flow, while the edge compression scheme can increase the throughput of the multiple flows only gradually. Therefore, the flow compression scheme can maintain a large number of compression processings as well as excellent throughput. On the other hand, with a large number of compression packets, the compression opportunity in the flow compression scheme is much smaller than that in the edge compression scheme. As a result, the edge compression scheme can obtain higher throughput than that of the flow compression scheme.

Next, we focus on the effect of compression time on the throughput performance. Figs. 6(a) and 6(b) respectively show the total throughput of each scheme when the compression time varies from 5 to 30 packets and when the number of compression packets of the proposed schemes is set to 5 (compression ratio: 0.6) or 10 (compression ratio: 0.5). The number of TCP flows is set to 198. The throughput of the flow and edge compression schemes increases as the compression time decreases. That is, the proposed schemes improve the performance as the processing speed on the edge nodes will have been higher. Similar to the results above, the flow compression scheme enables higher throughput than the other schemes with a small number of compression packets. However, the throughput of the flow compression scheme decreases as the compression time increases and is appropriately equal to that of the edge compression scheme with a large compression time. On the other hand, with a large number



(a) Throughput



(b) Number of compressing processings

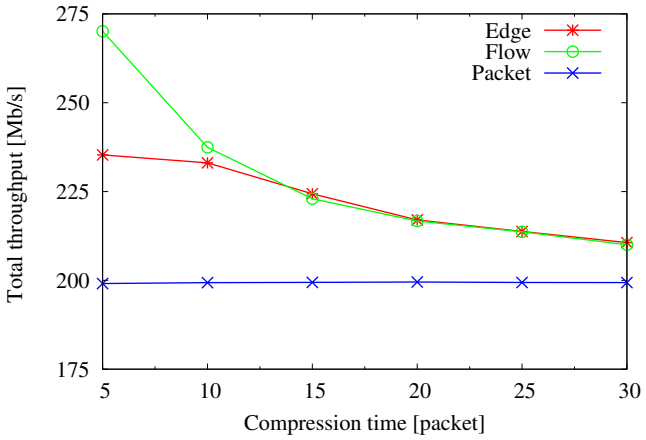
Figure 5. Effect of the number of compression packets

of compression packets, the edge compression scheme attains higher throughput than the other schemes over a wide range of compression time.

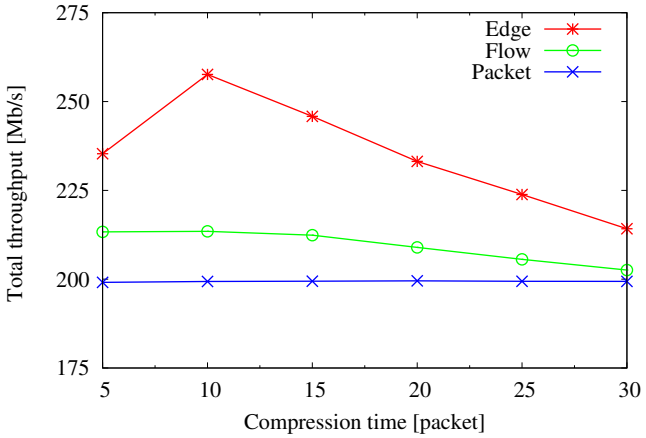
Figs. 7(a) and 7(b) respectively show the number of compression processings when the compression time varies from 5 to 30 packets and when the number of compression packets of the proposed schemes is set to 5 (compression ratio: 0.6) or 10 (compression ratio: 0.5). The number of TCP flows is set to 198. The number of compression processings of each scheme decreases as the compression time increases because a large compression time reduces the buffer capacity available to gather multiple packets having the same information as well as the compression opportunity. With a small number of compression packets, the flow compression scheme attains a larger number of compression processings than the edge compression scheme when the compression time is small. However, with a large compression time, there are no differences between the number of compression processings of each scheme. On the other hand, with a large number of compression packets, the number of compression processings of the edge compression scheme exceeds that of the flow compression scheme and is as large as that of the adaptive packet compression scheme with a large compression time.

These results demonstrate that the proposed schemes can

 ISBN: 978-1-61208-318-6

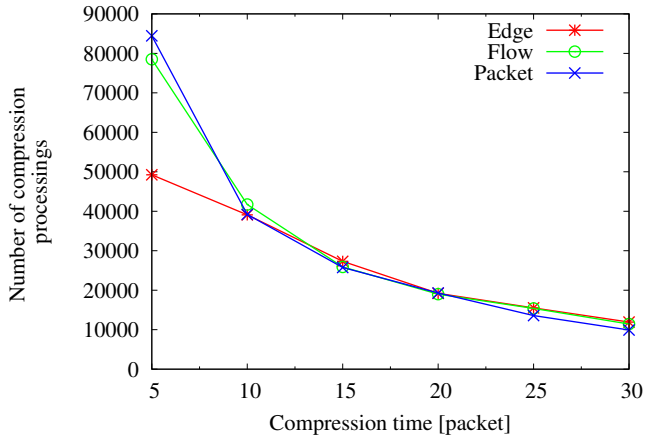


(a) Number of compression packets: 5

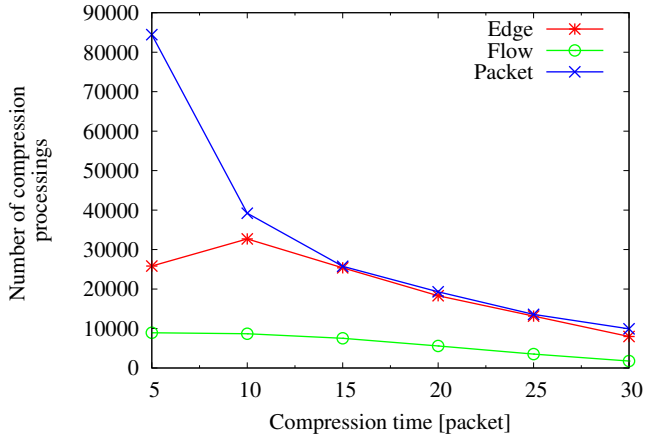


(b) Number of compression packets: 10

Figure 6. Effect of compression time: Throughput



(a) Number of compression packets: 5



(b) Number of compression packets: 10

Figure 7. Effect of compression time: Number of compression processings

improve the throughput by adaptively compressing multiple packets gathered in an output queue at edge nodes through utilization of the waiting time. The flow compression scheme enables high throughput with a small number of compression packets and a small compression time. Otherwise, the edge compression scheme enables higher throughput.

VI. Conclusion

To improve communication performance by efficiently decreasing Internet traffic, we have proposed adaptive online compression schemes that use flow information in advanced relay nodes. The proposed schemes gather adaptively multiple packets forwarded in the same direction in an output queue at advanced relay nodes and compress them by utilizing the waiting time. Through evaluations by simulation, we have shown that the proposed schemes enable high communication performance by compressing multiple packets. The flow compression scheme enables high throughput with a small number of compression packets and a small compression time. Otherwise, the edge compression scheme enables higher throughput. In our future work, we will design dynamic online compression algorithms that can adapt compression methods to network conditions and evaluate the proposed schemes using a prototype implementation from a viewpoint of the computational resources (processing time, memory usage, etc.).

Acknowledgment

This work was supported in part by the Japan Society for the Promotion of Science, Grant-in-Aid for Young Scientists (B) (No. 25730063).

References

[1] Cisco Systems, Inc., “Cisco visual networking index: Global mobile data traffic forecast update, 2012–2017,” Feb. 2013. Information available at http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white paper c11-520862.pdf. [Retrieved: Dec. 2013]

[2] M. Shimamura, H. Koga, T. Ikenaga, and M. Tsuru, “Compressing packets adaptively inside networks,” IEICE Transactions on Communications, vol. E93-B, no. 3, Mar. 2010, pp. 501–515.

[3] M. Shimamura, T. Ikenaga, and M. Tsuru, “A design and prototyping of in-network processing platform to enable adapting network service,” IEICE Transactions on Information and Systems, vol. E96-D, no. 2, Feb. 2013, pp. 238–248.

[4] S. Chen, S. Ranjan, and A Nucci, “IPzip: A stream-aware IP compression algorithm,” Proc. IEEE Data Compression Conference (DCC’08), Mar. 2008, pp. 182–191.

[5] The Network Simulator, http://www.isi.edu/nsnam/ns. [Retrieved: Dec. 2013]

 ISBN: 978-1-61208-318-6

Hybrid Synchrony Virtual Networks: Definition and Embedding

Rasha Hasan, Odorico Machado Mendizabal, Fernando Luís Dotti
Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil
Email: rasha.hasan@acad.pucrs.br, odoricomendizabal@furg.br, fernando.dotti@pucrs.br

***Abstract*—Virtual Networks (VNs) have attracted considerable attention in the last years since they offer a flexible and economic approach to deploy customer suited networks and run their applications. Such applications have different requirements, such as topology, security, resilience, and thus pose different challenges to the network embedding problem. In the last three decades of research in distributed systems, one core aspect discussed is the one of synchrony, since it impacts directly the complexity and functionality of fault-tolerant algorithms. In this paper, we argue that VNs and a suitable VN embedding process offer both abstractions and techniques to discuss and address the support of applications with hybrid synchrony demands, thus contributing in core aspects of reliable distributed systems. This work introduces the general idea of Hybrid Synchrony Virtual Networks (HSVNs) and presents a mathematical model that formalises the embedding of HSVNs into a physical network. Our results show that the model proposed is able to, correctly and efficiently, allocate resources on the SN, in an optimal manner.**

Keywords*—*Virtual Network; Distributed Systems; Synchrony

I. INTRODUCTION

Virtual Networks (VNs) have attracted considerable attention in the last years, both as an experimental environment to evaluate new protocols, as well as a technology to be integrated in the current network architectures. As can be seen in the literature, the diversity of applications pose different requirements on their supporting VNs, e.g., topology, security, and resilience requirements. In this context, network embedding, a key aspect that defines how resources of a physical network (also called Substrate Network - SN) are used to support VNs, assumes several variants according to the kinds of applications and respective VNs demands.

In the last three decades of research in Distributed Systems (DSs), also triggered by the impossibility result by Fischer, Lynch and Paterson in the 80's [1], one core aspect discussed is the one of synchrony. It is known that the development of DSs depends on the guarantees provided by the underlying infrastructure. If, on the one hand, infrastructures with synchronous guarantees contribute towards development of simpler and reliable systems, on the other hand providing such guarantees may be very expensive or even infeasible. Thus, the assumption of asynchronous environments was commonly adopted because both it is considered more realistic and any solution for the asynchronous case can be generalized to the synchronous case. Since dealing with the uncertainty inherent to asynchronous models requires complex algorithms, and due to evolution in networking technologies, more recently the assumption of partial synchrony has been considered in the literature [2], [3].

In this paper, we propose and argue that VNs and the VN embedding process offer both abstractions and techniques to support applications with Hybrid Synchrony (HS) demands (or partial synchrony). To the authors' best knowledge, this is undiscussed in the VN field and, as mentioned above, of paramount importance to host a prominent class of DSs. More specifically, the contributions of this paper are: (i) we introduce the need and the idea of VNs with hybrid synchrony requirements, characterise the kind of support needed from SNs to cope with these VN requirements, and thus formalise the main abstractions to discuss about hybrid synchrony both at VN and at SN level; (ii) we provide an example of an important distributed application, namely a failure detector, that benefits from hybrid synchronous infrastructures; (iii) we discuss and formalise the network embedding problem for VNs with hybrid synchrony requirements through a mathematical model; (iv) we evaluate the performance of our model in terms of mapping cost, physical resources load, embedding time, and measure the efficiency of our approach to spare synchronous resources.

The paper is organized as follows: Section II discusses related work. In Section III, we motivate the importance of hybrid synchrony to support DSs. In Section IV, we propose and formalize the notion of VNs with hybrid synchrony (HS), which we call HSVNs, together with the embedding model. Section V includes performance evaluation, and Section VI illustrates an example for the HSVN mapping. Finally, in Section VII , we conclude the paper.

II. RELATED WORK

Revising the literature, we found several works treating the VNs mapping problem through two main approaches: optimization models and heuristics. Chawdhury *et al.* [4] propose a relaxed version of mixed integer program, where the objective function is a weighted sum of node and link mapping, with the goal of increasing the acceptance ratio and decrease cost. Bay *et al.* [5] propose a security-aware mapping model where three levels of security are discussed. Yu *et al.* [6] propose an algorithm that combines VN mapping with substrate link backup to improve VNs resilience. Unlike the previous works, Hsu *et al.* [7] map virtual links through path splitting technique, in addition, path migration is used to maximize the number of coexisting VNs. Botero *et al.* [8] were the first to propose a heuristic algorithm that considers the CPU of the physical paths intermediate nodes. For wider

 ISBN: 978-1-61208-318-6

collection of VNs mapping, see Belbekkouche *et al.* [9], for survey.

In the tropic of VNs mapping, we find that our work is nearer to those who were concerned with delay constraints. For example, Zhang *et al.* [10] propose a heuristic algorithm for mapping virtual multicast service-oriented networks subject to delay and delay variation. They consider SNs composed of links with maximum delay. Their work benefits real-time and interactive applications, where packets are supposed to be received at the destination within specific time bounds, and the delay difference of packets reception at multiple destinations should be minimal.

Inführ *et al.* [11] addressed the VNs mapping problem with delay constraints besides routing and location constraints. The SN considered is composed of links with maximum delay, and nodes that have maximum routing capacity and location constraints. Four different categories were used to represent cases in which VNs have different sets of requirements regarding BW, delay, and nodes CPU: *(1) web slice* for low BW requirements, short delays, and no specific CPU requirements, *(2) stream slice* for medium to high BW requirements, no delay bounds, and 3 processing units per routed bandwidth, *(3) P2P slice* for medium BW and CPU requirements and no delay bounds, and *(4) VoIP slice* for medium BW and delay requirements, and high CPU requirements.

The study presented in this paper is distinct from the aforementioned works in the following aspects: *(1)* we consider the delay constraints (or time bounds) on both links and nodes, not only links, since in the considered class of DSs some links should have time guarantees in delivering the messages, and some nodes should be performing real-time tasks; *(2)* a physical path is considered synchronous not only when its links are synchronous, rather the path's intermediate nodes should be all synchronous as well, since they play role in the routing process, impacting the source-destination delay; *(3)* the mapping model we propose aims at optimizing the usage of the synchronous resources whose building cost is high comparatively. For example, some VNs slices adopted in [11] had no delay requirements, yet the SN considered had no distinction in kind of resources, which results in an unneeded cost, and *(4)* unlike other works, the SN we consider is hybrid in its components synchrony. Some nodes and links have time bounds and others do not, which is suitable for DSs applications that have hybrid synchronous requirements.

III. Work motivation: Synchrony Models in distributed systems

The design of DSs is strongly dependent on the assumptions about the environment where they execute. For instance, different assumptions about process execution speeds and message delivery delays would require specific design decisions. Thus, an important aspect to consider is the synchrony level offered by the underlying infrastructure. In an asynchronous system, no assumption about process execution speed and/or message delivery delays is made. Conversely, in a synchronous system, relative processing speed and the message delays are bounded [12].

Assuming that, underlying infrastructures behaving asynchronously showed to be realistic to a wide range of applications. Although they are very attractive, key problems of fault-tolerant computing are not solvable under the asynchronous assumption. For example, Fischer, Lynch and Paterson have proven that consensus cannot be solved deterministically in asynchronous systems where at least one process may crash [1].

By asserting that a system is synchronous, system developers can rely on the timely behaviour of the components. This, in turn, enables one to employ simpler algorithms than those required to solve the same problem in an asynchronous system [12]. For instance, processes can perfectly distinguish faulty from slow processes. However, building synchronous systems requires infrastructures composed exclusively by timely components, which could be very expensive or even infeasible.

Hybrid models assume intermediate levels of synchrony. Cristian and Fetzer proposed the timed-asynchronous model [2], where the system alternates between synchronous and asynchronous behaviour. In that model, the degree of synchronism varies over time. In [3], Veríssimo presented the wormhole model, that exploits the space dimension to provide hybrid synchrony. This means that timely guarantees of system components may be different. For instance, one part of a system would behave synchronously, while other part would be fully asynchronous.

Once behaviours caused by faults and arbitrary delays are expected in the conventional infrastructures, hybrid models become a good option to improve the development of fault-tolerant applications. By enforcing small parts of the system to behave synchronously while other parts are asynchronous, stronger properties provided by synchronous parts can be enjoyed by the system as a whole. For this reason, hybrid systems overcome limitations of the homogeneous systems.

Example: Failure detector - Failure detectors have attracted interest in the development of reliable DSs, since consensus and related problems (e.g., atomic broadcast [13]) can be solved with it. The failure detection approach can also be adapted to solve other relevant problems, such as predicate detection [14] and election [15].

Failure detectors are used to detect faulty processes in a group of processes, and they are defined in terms of abstract properties, namely *accuracy* and *completeness*. A failure detector that satisfies *strong accuracy* and *strong completeness* properties is a *perfect failure detector* ($\mathcal{P}$) [13]. It means it never makes mistakes (suspect erroneously) and, eventually detects every crash.

A failure detector $\mathcal{P}$ can be built on top of synchronous environments. The problem is that implementing $\mathcal{P}$ in fully synchronous environments depends on the existence of an underlying infrastructure with timely guarantees (sometimes infeasible) while implementing it in asynchronous systems is even impossible.

Macêdo *et al.* [16] propose an implementation of a failure detector $\mathcal{P}$ that runs on hybrid synchronous environments. They assume the underlying system has synchronous processes, some channels behave synchronously and others asynchronously.

Basically, each module fd_i periodically asks to processes p_j if they are alive. Upon receiving a message "are you alive",

 ISBN: 978-1-61208-318-6

every correct process replies to the sender with a "I'm alive" message. Upon receiving the replying message, fd_i knows the process p_j is up. However, if a timeout expires, it means that no answer from p_j was received in the last τ time units. If the channel connecting processes fd_i to fd_j is synchronous, then it is known that the process p_j has failed. Process p_j is added to the faulty list in p_i, and a notification informing the detection is sent to all other processes. Otherwise, if the channel is asynchronous, there is no way to detect if the process p_j has failed or the reply message is delayed.

We illustrate a failure detector $\mathcal{P}$ running in a hybrid synchronous environment in Figure 1. It shows a hypothetical topology for an application composed by six processes. All processes are hosted in synchronous nodes, and they communicate with each other through payload channels (pa_i). Further, a failure detector module fd_i is attached to each process P_i. Connection between failure detectors modules in a synchronous partition is done by synchronous channels (solid lines in the figure). Connection between fd modules in different partitions can be asynchronous (dotted lines). In order to improve legibility, payload channels pa_i, CPU and bandwidth constraints were omitted in the figure. In this example, the payload channels should be represented by a complete graph connecting every pair of processes.

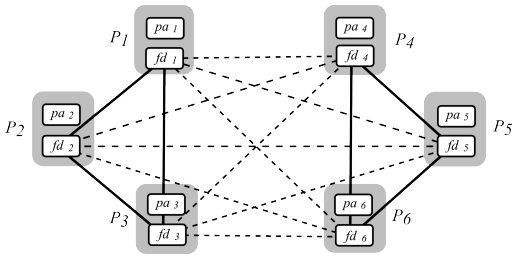


Fig. 1: Application topology with failure detector $\mathcal{P}$

Although not all failure detectors are in the same synchronous partition, the $\mathcal{P}$ implementation allows every application process to benefit from a perfect detection. Even in cases in which not all fd_i modules belong to a synchronous partition, it is possible to take advantage of the existing synchrony, provided that some subgraphs are synchronous. In such cases, assumptions from weaker failure detectors (e.g., $\Diamond\mathcal{P}, \Diamond\mathcal{S}$ [13]) would be ensured and still useful for the applications.

Another interesting aspect of the hybrid synchronous system is that application workload is totally independent of the failure detector modules. Application processes can communicate through asynchronous channels and still enjoy stronger properties provided by the failure detector service.

IV. Our proposal: Hybrid Synchrony Virtual Networks

Our proposal lays in offering VNs to support DSs applications with partial synchrony. The rational behind it is mainly twofold. First; the synchronous elements in DSs are considerably more expensive than the asynchronous ones, since they require fundamental handling mechanisms. Resource sharing, provided by the nature of VNs, allows simultaneous use for this class of expensive resources. Secondly, VNs allow flexible resource allocation mechanisms by the Infrastructure Service Provider (ISP). This provides modern DSs applications with scalability which is an important aspect in the field.

The aforementioned reasons lead to the abstraction of new type of VNs: the Hybrid Synchrony Virtual Networks (HSVN). They are virtual networks that have a subset of nodes and links that obey time bounds for processing and communication. This abstraction put us to meet two main aspects associated: (i) the SN design, since VNs inherit properties that only exist in the underlying infrastructure, and (ii) suitable efficient embedding process for the HSVN.

Although HSVN can run on fully synchronous SN, this decision would have to pay the excess in an unneeded cost, since even asynchronous virtual nodes and links will be mapped on synchronous physical ones. We argue that hybrid synchronous SN, combined with a suitable embedding, is capable to answer the timely requirements in an economic manner. Hybrid synchronous SNs have two classes of nodes: (i) *synchronous nodes* with functioning time guarantees, achieved through the implementation of periodical real-time tasks, and (ii) *asynchronous nodes* that have no timely guarantees. Analogously, two classes of physical links are available: (i) *synchronous links* that have time-bounded messages transmission delay, achieved through the implementation of Quality of Service (QoS) policies and admission control, and (ii) *asynchronous links* that have no timely guarantee.

A. The HSVN embedding model

We propose a HSVN embedding mathematical model in the shape of a Mixed Integer Program (MIP). Our model answers hybrid synchrony requirements and, at the same time, optimizes the synchronous resources usage besides the BW and CPU. The HSVN exploits the space dimension to provide hybrid synchrony [3].

Variables definition- The substrate network is represented by an undirected graph $G(N, L)$, composed of a set of physical nodes N connected through a set of physical links L. N is given by $N_s \cup N_a$, where N_s and N_a contain all the synchronous and asynchronous SN nodes, respectively. Similarly, L is given by $L_s \cup L_a$. Each virtual network VN^k belonging to the set of virtual networks VN will be presented by an undirected graph $G^k(N^k, L^k)$, where $N^k = N^k_s \cup N^k_a$ and $L^k = L^k_s \cup L^k_a$. We consider that there is a cost $c(i, j)$ for one unit of traffic going through the physical link $(i, j) \in L$. Analogously, $c(i)$ is the cost for processing one unit of traffic in node $i \in N$. $c(i, j)$ and $c(i)$ are of higher value if the link and node were synchronous. A binary function $sync(i)$ expresses the SN nodes synchrony: $sync(i) = 1$ if $i \in N_s$, otherwise $sync(i) = 0$ (i.e., $i \in N_a$). Similarly, $sync(i, j)$ expresses the SN links synchrony. Functions $sync(i^k)$ and $sync(i^k, j^k)$ indicate the virtual nodes and links synchrony respectively ($i^k \in N^k$ and $(i^k, j^k) \in L^k$). Besides synchrony, two other attributes are considered for the SN and VN elements: nodes CPU, and links bandwidth (BW). The syntax for those attributes on the SN and VN respectively are: $cpu(i)$, $bw(i, j)$, $cpu(i^k)$, and $bw(i^k, j^k)$. Finally, we define the output variables for our mathematical model: a binary function $\sigma(i^k, i)$ that expresses whether node $i \in N$ maps node $i^k \in N^k$, and a binary function $\rho(i^k, j^k, i, j)$ that expresses whether the

physical link $(i,j) \in L$ is part of the path that maps the virtual link $(i^k, j^k) \in L^k$.

The embedding model- The Objective Function (O.F.) we consider is inspired from [10], which is the total resources used (e.g., BW and CPU). We modify the O.F. with the goal of minimizing the use of synchronous resources besides the BW and CPU. For this purpose, $c(i)$ and $c(i,j)$ are inserted in (1).

Objective: minimize

$$\sum_{\forall VN^k \in VN} \sum_{\forall (i^k) \in N^k} \sum_{\forall (i) \in N} (\sigma(i^k, i) \cdot c(i) \cdot cpu(i^k)) + \sum_{\forall VN^k \in VN} \sum_{\forall (i^k, j^k) \in L^k} \sum_{\forall (i,j) \in L} (\rho(i^k, j^k, i, j) \cdot c(i,j) \cdot bw(i^k, j^k)) \quad (1)$$

Subject to

- *Capacity constraints:*
for every $(i,j) \in L$

$$\sum_{\forall VN^k \in VN} \sum_{\forall (i^k, j^k) \in L^k} \rho(i^k, j^k, i, j) \cdot bw(i^k, j^k) \leq bw(i,j) \quad (2)$$

for every $i \in N$

$$\sum_{\forall VN^k \in VN} \sum_{\forall i^k \in N^k} \sigma(i^k, i) \cdot cpu(i^k) \leq cpu(i) \quad (3)$$

- *Nodes mapping constraints:*
for every $VN^k \in VN$, $i^k \in N^k$

$$\sum_{\forall i \in N} \sigma(i^k, i) = 1 \quad (4)$$

for every $VN^k \in VN$, $i \in N$

$$\sum_{\forall i^k \in N^k} \sigma(i^k, i) \leq 1 \quad (5)$$

- *Links mapping constraint:*
for every $VN^k \in VN$, $(i^k, j^k) \in L^k$, $i \in N$

$$\sum_{\forall j \in N} \rho(i^k, j^k, i, j) - \sum_{\forall j \in N} \rho(i^k, j^k, j, i) = \sigma(i^k, i) - \sigma(j^k, i) \quad (6)$$

- *Nodes synchrony constraints:*
for every $VN^k \in VN$, $i^k \in N^k$, $i \in N$

$$sync(i^k) \cdot \sigma(i^k, i) \leq sync(i) \quad (7)$$

- *Links synchrony constraints:*
for every $VN^k \in VN$, $(i^k, j^k) \in L^k$, $(i,j) \in L$

$$sync(i^k, j^k) \cdot \rho(i^k, j^k, i, j) \leq sync(i,j); \quad (8)$$

for every $VN^k \in VN$, $(i^k, j^k) \in L^k$, $(i,j) \in L$

$$sync(i^k, j^k) \cdot \rho(i^k, j^k, i, j) \leq sync(i) * sync(j); \quad (9)$$

The capacity constraint (2) assures that the total bandwidth of the virtual links, mapped on paths that include a certain physical link, does not exceed the bandwidth capacity of this physical link. Similarly, constraint (3) represents the equivalent restriction regarding nodes CPU. The node mapping constraint (4) assures that each virtual node is mapped, and only once, on a physical node. Without this constraint, and since the O.F. aims at minimizing cost, then the optimizer might choose not to map any node, which is against the goal. Constraint (5) assures that virtual nodes belonging to the same VN are not mapped on the same physical node. This is to achieve load balancing besides improving the reliability, since the unavailability of a SN node will impact, at most, one node on a given VN. This procedure minimizes the number of virtual nodes prone to failure by a physical node failure. This is an important aspect in fault tolerant distributed systems, where a maximum number of faulty processes is accepted to allow the system to tolerate the fault, i.e., not to let the fault impact the system output. For any virtual link (a,b), the links mapping constraint (6), adopted in [5] and [11], assures the creation of a valid physical path. Because the right side of the equation will be 1 and -1 for a and b respectively, meaning a will have an outgoing arc and b an ingoing one. For all other nodes on the SN, the right side of the equation will be zero, thus the concatenation of arcs will form a valid path. The nodes synchrony constraint (7) assures that synchronous virtual nodes are mapped only on synchronous SN nodes, whereas asynchronous virtual nodes are allowed to be mapped on synchronous or asynchronous SN nodes. This is acceptable because the synchronous SN nodes supply what the asynchronous ones do, but the reverse is not valid. Similarly, the links synchrony constraint is presented in (8). Note that the allocation of synchronous physical resources for asynchronous virtual demands is done only if there are no other possible options (physical asynchronous resources got exhausted). This is achieved via minimizing the O.F. Finally, constraint (9) guarantees that when the intermediate physical nodes on the synchronous physical path should be also synchronous. This is because these nodes play role in the routing process, thus impacting the source-destination delay. After solving the mathematical model, each virtual node is mapped to one physical node, and each virtual link is mapped to one physical path at maximum, where a physical path can be a unique physical link or a concatenation of physical links.

V. PERFORMANCE EVALUATION

We evaluate the performance of our model through: 1) mapping cost, 2) physical resources load, 3) optimizing the usage of synchronous resources and, 4) embedding time.

A. Workloads and tools

Like some other works [17] [5], the physical and virtual networks were randomly generated. For this we used BRITE [18] tool (Boston university Representative Internet Topology gEnerator) with Waxman [19] model. We implemented the model with ZIMPL language [20] (Zuse Institute Mathematical Programming Language) and used CPLEX Optimization Studio [21] to solve the MIP, running on a computer with a CPU of 4 cores and 1.60 GHz, and 2 GB of main memory. We ran twelve experiments divided into three groups, A, B and C, with VNs total size of 10, 20, and 30 nodes respectively. Table I describes the parameters for each experiment.

 ISBN: 978-1-61208-318-6

TABLE I: Experimentsparameters

Group:VN size	A:10 routers,B:20 routers,C:30 routers			
Scenario	**1**	**2**	**3**	**4**
SN size	25 nodes			
SN BW	uniformly distributed: 1Gbps-3Gbps			
SN CPU	nodes fully free initially			
VNs BW	uniformly distributed: 100Mbps-1Gbps			
VN CPU	10, 15, 25 % of SN nodes CPU			
SN sync.	30%			100%
VNs sync.	0%.	30%	60%	x%

In all the experiments, the SN size was fixed in 25 nodes. Initially, all CPUs are free, and links BW is uniformly distributed between 1-3 Gbps. In scenarios 1, 2, and 3 of each group, the SN was set up with 30% of synchronous resources, whereas in scenario 4 of each group the SN was fully synchronous. This scenario will be the base for cost comparison since it simulates the case where all the SN nodes and links have time bounds.

The VNs were generated with 3, 4, or 5 nodes each, the virtual nodes have 10%, 15%, or 25% of the SN nodes CPU. The VNs links BW was uniformly distributed between 100 Mbps and 1 Gbps. The VNs synchrony varies within each group: 0% in scenario 1, 30% in scenario 2 and 60% in scenario 3. Note that the VNs synchrony in the fourth scenario of each group was referred to as x% because in this scenario the mapping cost will be independent of the VNs synchrony requests since the SN resources have no differentiation in synchrony (the SN is fully synchronous).

B. Results

The first parameter evaluated is the mapping cost, represented by our model objective function This parameter is a combination of CPU and BW used. Figure 2 depicts the mapping cost for each of the twelve experiments performed. We note down three main observations: *(i)* within each group, the mapping cost increases gradually with the increment of the VNs synchrony requests. For example, the mapping cost increased 173% when the VNs synchrony demands increased from 0% in B1 to 30% in B2, and increased more 76% in B3 with 60% VNs synchrony demands. This is explained by the increase in physical synchronous resources (nodes and links) usage, which are more expensive. *(ii)* by comparing the counterparts experiments of the three groups, e.g., A2, B2, and C2 (all with 30% VNs synchronous demands), we notice that the mapping cost increases. This is due to the incremental VN size, which tends naturally to reserve more physical resources. *(iii)* comparing the three first experiments within each group with the fourth one, we can say that our model can host hybrid VNs in an economic way. That is, the SN can be used in an optimized way to allocate these demands. For instance, experiment C3 depicts the mapping of a hybrid VN with 60% of synchrony demands on a hybrid SN with 30% of synchronous resources. Whereas mapping the same VNs demand on a fully synchronous SN, experiment C4, is subject to an extra 94% un-needed cost. So, hybrid VNs do not need fully synchronous SN, rather a hybrid SN with suitable mapping is enough to allocate the needed demands, and spares resources for future ones.

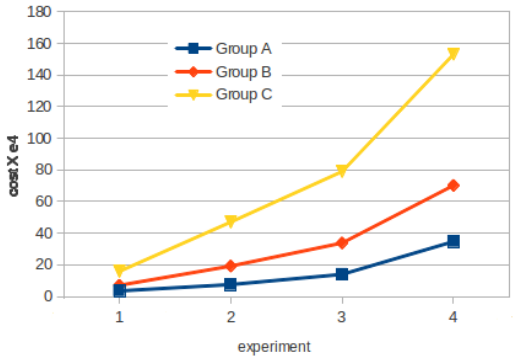


Fig. 2: Mapping cost

The second parameter to evaluate is the SN resources load. We present the load evaluation for scenario C3 only, because, within the twelve experiments performed, it is the one with the maximum VNs size and higher synchrony percentage as hybrid substrate. Figure 3 shows the cumulative distribution function (CDF) of the SN nodes CPU and links BW usage. We note that only 4% of the SN nodes reached 65% of use, and about 3% of the SN links have BW consumption that exceeds 60%. So, the SN resources seem to have fair load, which is an important factor since avoiding to fully charge nodes and links tends increase the possibility of mapping future demands within the same SN.

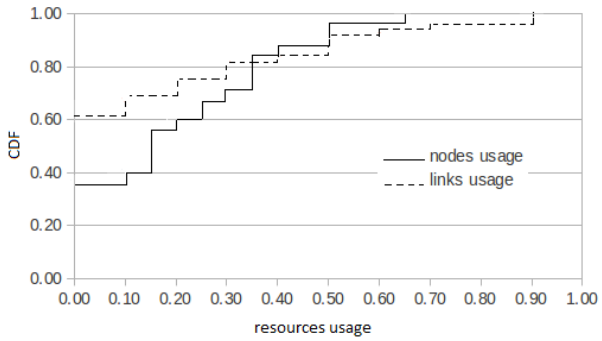


Fig. 3: CDF for resource usage in experiment C3

Next, to check closely the model ability in sparing the synchronous resources, we devised a scenario where all asynchronous SN resources get used. Consequently, synchronous SN resources are used for mapping asynchronous VNs demands. This case is allowed only when SN is running out of asynchronous physical resources. To investigate this point, we modified experiment B2 in the set to experiment B2', where the SN in B2' has 25% asynchronous resources and the VNs CPU demands increased to 60% of the SN nodes CPU. Figure 4 is a scheme of resource mapping in B2'. The horizontal axis is the SN resources, divided into synchronous in the positive portion of the axis, and asynchronous in the negative one. The same for the VNs synchronous and asynchronous demands on the vertical axis. Note that, the altitude holds no information, it is just the positioning (positive or negative). This division results in four quarters, we number them counter clockwise. Optimally, the synchronous demands are to be

 ISBN: 978-1-61208-318-6

mapped on synchronous SN resources, and asynchronous on asynchronous. This leads to points allocated only in the first and third quarters. No points are supposed to appear in the second quarter, since it is meaningless to map synchronous virtual demands on asynchronous physical resources. The fourth quarter is supposed to have the minimum number of allocations, which is an indication of optimizing the use of synchronous resources (i.e., few mappings of asynchronous demands on synchronous physical resources). In other works, that consider fully synchronous SN, all the allocations will appear in the right half of the plane (quarter 1 and 4), which is the expensive part. In our work we insert the possibility of allocations existing in the left half of the plane, which reduces the cost potential.

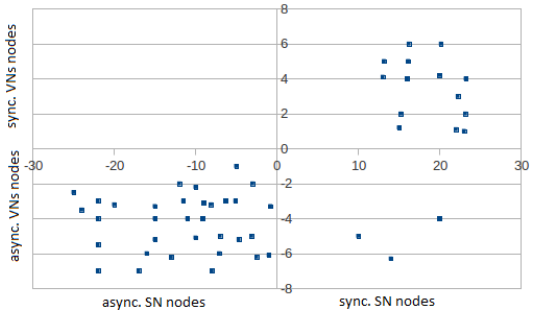


Fig. 4: Scheme of resource mapping in experiment B2'

Finally, We evaluate the optimization time for each experiment performed, see Table II. We notice that most of the values were less than 10 minutes which is a reasonable computational time. For some experiments the value was high, which might be a compromise for obtaining an optimal solution.

TABLE II: Embedding time (in minutes)

Group.	exp.1	exp.2	exp.3
A	0.85	0.40	0.31
B	37.55	1.64	10.26
C	58.34	27.12	4.47

VI. Graph-based example for HSVN mapping

In this section, we present a simple graph-based example for mapping HSVNs. The goal is to see, in practice, some aspects considered by our approach.

We consider three VNs with different synchrony demands, together with a hybrid SN. Both the VNs and the SN are shown in Figure 5. VN^1 represents a virtual infrastructure for an application equipped with a failure detector, similar to that presented in Section III, but with four nodes. The links connecting payload channels are omitted in the figure to improve legibility. VN^1 has hybrid requirements regarding synchrony, whereas VN^2 and VN^3 represent fully synchronous and asynchronous applications, respectively. By solving the MIP for the example under analysis, every virtual node was mapped on one physical node, and each virtual link was mapped on one physical path, where a physical path can be one physical link, or a concatenation of several physical links. Figure 5 shows the optimal solution found for nodes and links mapping.

In the light of this example, we point at some aspects previously detailed in the paper: *i)* our mapping approach considers both hybrid VNs (e.g., VN^1) and homogeneous VNs (e.g., VN^2 and VN^3), *ii)* the proposed model aims at sparing the synchronous resources, e.g., an asynchronous virtual link (fd_2^1, fd_4^1), was mapped on a path of three asynchronous links {(b,h),(h,a),(a,c)}, connecting b to c, to avoid mapping it on a synchronous shorter path of one link {(b,c)}, connecting the same two nodes, *iii)* the HSVN allows resource sharing, which is important, especially for sharing the synchronous resources, e.g., the synchronous nodes c, and d could be used both for mapping VN^1 and VN^2. The same with the synchronous link (d, c), finally, *iv)* regarding mapping cost, we ran the optimizer for the adopted example in two cases, first, with a hybrid SN as illustrated in Figure 5, secondly, with a fully synchronous SN. The O.F. values obtained were 5400 and 11400 respectively in both cases. This shows clearly that, the use of hybrid SN, together with a suitable mapping process, minimizes the cost considerably.

VII. Discussion

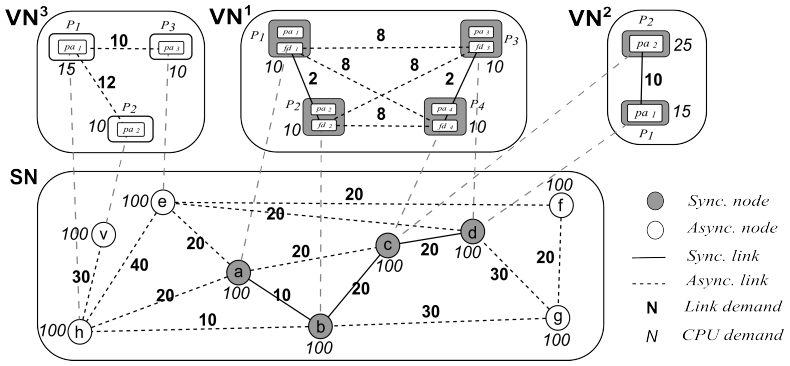
In this paper, we have proposed the concept of Hybrid Synchrony Virtual Networks, i.e., virtual networks that have a subset of nodes and links that obey time bounds for processing and communication. The rationale behind it is, at one hand, that there is an important class of systems, namely fault-tolerant distributed systems, that can benefit from the hybrid synchrony. On the other hand, the embedding of several virtual networks in a substrate network allows resource sharing, which is important since synchronous resources are expensive. The proposed embedding mathematical model adopts these aspects.

Our results show that our model can host hybrid synchrony VNs in an economical way, which is achieved mainly through: 1) the usage of a hybrid synchronous SN instead of a fully synchronous one, and 2) synchronous resources are spared, in other words, mapping asynchronous virtual demands on top of synchronous physical resources is considered the last resource invested only before rejecting the demand. Moreover, the model reflects a reasonable load distribution on the underlying nodes and links, and acceptable optimization time.

Our work can be generalized in three directions: *(i)* although we have dedicated enough efforts to illustrate perfect failure detectors, a wider set of applications benefit from hybrid synchrony. For instance, general purpose applications would communicate mainly through asynchronous channels and still rely on timely execution triggers. Thus, certain actions would be executed in a timely fashion (e.g., checkpointing [22], election [15], or any round-based agreement); *(ii)* the hybrid SN we are proposing, combined with our embedding model, can host not only hybrid synchrony applications, but also homogeneous ones (fully synchronous or fully asynchronous); *(iii)* while in this step of our work we are concerned with synchrony, we envisage that similar models may, in the future, be used to denote other kinds of specific functionalities expected from the resources. such as subsets of nodes and links with special security or resilience features.

Our future work goes in the direction of online mapping for the HSVNs, when the SN resources, or/and the VNs demands are time variant. The mapping approach proposed

 ISBN: 978-1-61208-318-6



VN^1 **links**	**SN path**
(fd_1, fd_2)	$\{(a,b)\}$
(fd_1, fd_3)	$\{(a,e),(e,d)\}$
(fd_1, fd_4)	$\{(a,c)\}$
(fd_2, fd_3)	$\{(b,g),(g,d)\}$
(fd_2, fd_4)	$\{(b,h),(h,a),(a,c)\}$
(fd_3, fd_4)	$\{(d,c)\}$

VN^2 **links**	**SN path**
(pa_1, pa_2)	$\{(d,c)\}$

VN^3 **links**	**SN path**
(pa_1, pa_2)	$\{(h,v)\}$
(pa_1, pa_3)	$\{(h,e)\}$

Fig. 5: Virtual networks mapping

in this paper, allows only static resource allocation, and thus, does not answer the online mapping. For this reason, we are developing a heuristic algorithm, which is supposed to allow resource allocation for the HSVNs in a dynamic manner.

ACKNOWLEDGEMENT

This work has been supported by FAPERGS-NPRV (Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul - Nucleo de Pesquisa em Redes Virtuais), project PRONEM 11/2038-1.

REFERENCES

[1] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process," *Journal of the ACM*, vol. 32, no. 2, pp. 374–382, 1985.

[2] F. Cristian and C. Fetzer, "The timed asynchronous distributed system model," *IEEE Trans. on Parallel and Distributed Systems*, vol. 10, no. 6, pp. 642–657, 1999.

[3] P. E. Veríssimo, "Travelling through wormholes: a new look at distributed systems models," *ACM SIGACT News*, vol. 37, no. 1, pp. 66–81, 2006.

[4] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

[5] L. R. Bays, R. R. Oliveira, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Security-aware optimal resource allocation for virtual network embedding," in *Proc. of CNSM*, 2012.

[6] Y. Yu, C. S. zhi, L. Xin, and W. Yan, "Rmap: An algorithm of virtual networks resilience mapping," in *Proc. of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2011.

[7] W. H. Hsu, Y. P. Shieh, C. H. Wang, and S. C. Yeh, "Virtual network mapping through path splitting and migration," in *Proc. of The 26th International Conference on Advanced Information Networking and Applications Workshops*, 2012.

[8] J. F. Botero, X. Hesselbach, A. Fischer, and H. De Meer, "Optimal mapping of virtual networks with hidden hops," *Telecommunication System*, vol. 52, no. 3, pp. 1–10, 2013.

[9] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Communication Surveys and Tutorials*, vol. 14, no. 4, pp. 1114–1128, 2012.

[10] M. Zhang, C. Wu, M. Jiang, and Q. Yang, "Mapping multicast service-oriented virtual networks with delay and delay variation constraints," in *IEEE GLOBECOM*. IEEE Communication Society, 2010.

[11] J. Infuhr and G. R. Raidl, "Introducing the virtual network mapping problem with delay, routing and location constraints," in *Proc. of 5th International Networking Optimization Conference (INOC)*.

[12] F. B. Schneider, "Distributed systems (2nd ed.)," S. Mullender, Ed. ACM Press/Addison-Wesley Publishing Co., 1993, ch. What good are models and what models are good?

[13] T. D. Chandra and S. Toueg, "Unreliable failure detectors for reliable distributed systems," *Journal of the ACM*, vol. 43, no. 2, pp. 225–267, 1996.

[14] F. C. Gartner and S. Kloppenburg, "Consistent detection of global predicates under a weak fault assumption," in *The 19th IEEE Symposium on Reliable Distributed Systems*. IEEE, 2000.

[15] H. Matsui, M. Inoue, T. Masuzawa, and H. Fujiwara, "Fault-tolerant and self-stabilizing protocols using an unreliable failure detector," *IEICE Trans. on Information and Systems*, vol. 83, no. 10, pp. 1831–1840, 2000.

[16] R. de Araujo Macedo and S. Gorender, "Perfect failure detection in the partitioned synchronous distributed system model," in *Proc. of International Conf. on Availability, Reliability and Security (ARES)*, 2009.

[17] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM-SIGCOMM*, vol. 38, no. 2, pp. 17–29, 2008.

[18] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: Boston university representative internet topology generator." [Online]. Available: http://www.cs.bu.edu/brite

[19] B. M. Waxman, "Routing of multipoint connections," *Selected Areas in Communications*, vol. 6, no. 9, pp. 1622–1617, 1988.

[20] T. Koch, "Rapid mathematical programming," Ph.D. dissertation, Tichnische Universität Berlin, 2004.

[21] IBM, "Cplex." [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer

[22] E. N. Elnozahy, L. Alvisi, Y.-M. Wang, and D. B. Johnson, "A survey of rollback-recovery protocols in message-passing systems," *ACM Computing Surveys (CSUR)*, vol. 34, no. 3, pp. 375–408, 2002.

 ISBN: 978-1-61208-318-6

Modeling of Content Dissemination Networks on Multiplexed Caching Hierarchies

Satoshi Imai
Fujitsu Laboratories Ltd.
4–1–1 Kamikodanaka, Kawasaki,
Kanagawa, Japan
Email: imai.satoshi@jp.fujitsu.com

Kenji Leibnitz
CiNet, NICT and Osaka University
1–4 Yamadaoka, Suita,
Osaka, Japan
Email: leibnitz@nict.go.jp

Masayuki Murata
Osaka University
1–5 Yamadaoka, Suita,
Osaka, Japan
Email: murata@ist.osaka-u.ac.jp

***Abstract*—In-network caching technologies like *Content-Centric Networking* (CCN) are expected to reduce the network traffic and improve the service quality, such as communication latency, by storing content data on routers near to users. Meanwhile, the adaptive cache management using *Time-To-Live* (TTL) of content can realize efficient memory management per content. However, for a distributed cache system such as CCN, it is difficult to evaluate cache performance and network resources required in the cache mechanism using the TTL value. Therefore, we propose a theoretical model, which can analyze the impact of TTL-based caching on network resources and cache performance, and evaluate some scenarios using the proposed model. We finally introduce a cache mechanism using energy efficient TTLs and show its effectiveness by the model-based analysis.**

Keywords-In-network caching; distributed cache system; Content Centric Networking; Time-To-Live

I. INTRODUCTION

The currently increasing network traffic is caused by the growing number of content dissemination services in the network and *Content Delivery Networks* (CDN) are well known as efficient content delivery mechanisms. Since the CDN service can provide content delivery at the edge of the networks by allocating content replicas in cache servers, which are in geographical proximity to users, it is expected to reduce the network traffic. Moreover, the caching services can improve communication quality, such as latency and throughput, in the delivery of content. Recently, a new communication paradigm, namely *Content Centric Networking* (CCN) [1], has been proposed. The CCN-enabled routers have autonomous caching functionality for content data. In the content dissemination mechanism of CCN, content publishers advertise newly released content from the origin site of the content along predefined routes. A content request (*Interest*) is forwarded on each content router (*CR*) based on the *Forwarding Information Base* (FIB) until the requested content is found. An *Interest* forwarded on a *CR* is added to the *Pending Interest Table* (PIT) in order to remember the interface on which to send back the replies (*Data*). When the requested content is found on a *CR*, *Data* of the content are transmitted based on the PIT. Furthermore, *Data* are cached on all *CR*s along the transmission route based on the specific replacement policy such as Least Recently Used (LRU) or Least Frequently Used (LFU). Therefore, CCN can automate the placement and delivery of *Data* by *CR*s on networks and it is highly expected to reduce network traffic and improve the communication quality.

However, the network traffic and communication quality are influenced by the cache locations because content is generated by many publishers at various locations in CCN. Moreover, the caching performance depends on the memory size in each *CR* on multiplexed delivery trees rooted at each origin site of content. Therefore, it is a major issue to analyze the impact of the distributed cache mechanism on network resources, such as memory or network devices, and cache performance, such as cache hit ratio and hop length, while delivering content.

Meanwhile, the persistent storage of content in each *CR* is inefficient because the content, such as video streaming, generally has a limited lifetime. Therefore, dynamic caching mechanisms often use a limited period of time, called *Time-To-Live* (TTL) for content. Moreover, TTL-based caching can improve the scalability of cache management compared with LRU or LFU replacement with cache coordination which sorts content by popularity and selects which content should be discarded. In this paper, we first propose an analytical model to evaluate the impact of TTL on cache performance and network resources of content in a distributed cache mechanism like CCN. Furthermore, we demonstrate evaluation results for some network scenarios using the proposed model. In addition, we introduce a cache mechanism using energy efficient TTLs as one possible application of TTL-based caching and show the effectiveness of the proposed cache mechanism using our model.

The remainder of this paper is organized as follows. Section II discusses the TTL-based cache mechanism followed by Section III which summarizes related work. We propose our analytical model in Section IV and demonstrate evaluation results using the proposed model in Section V. Furthermore, in Section VI we introduce a cache mechanism to improve energy efficiency using TTL and we evaluate the effectiveness of the proposed mechanism. Finally, we conclude the paper in Section VII.

II. TTL-BASED CACHE MECHANISM AND ISSUES

In this paper, we assume that each *CR* executes data caching using TTL of content which can, for instance, be signaled in the data header of the content or set at each *CR* in advance.

In TTL-based caching, each *CR* resets the time counter to

 ISBN: 978-1-61208-318-6

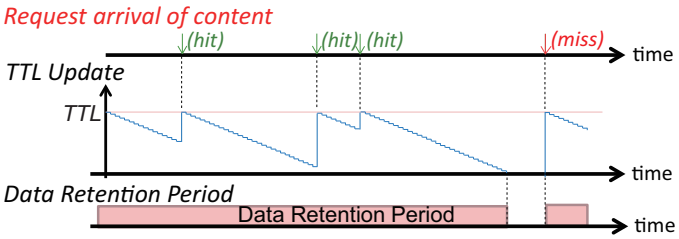


Figure 1: Traditional TTL-based caching

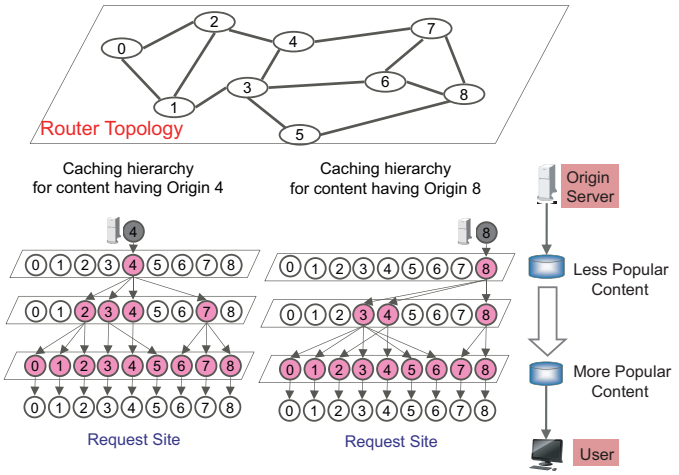


Figure 2: An example of caching hierarchies for origin sites 4 and 8

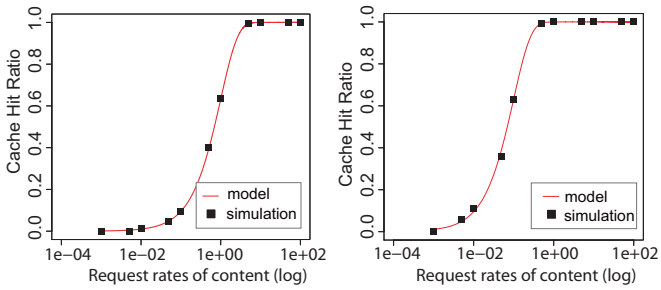
the TTL of content every time a new request for this content arrives and decreases the counter by 1 every time unit (cf. Figure 1). In this mechanism, each *CR* caches data of content delivered by another *CR* or an origin server when the content counter is above 0 and discards the data of content when the counter becomes 0.

Meanwhile in CCN, each *CR* autonomously constructs some caching hierarchies rooted at each origin site of content (cf. Figure 2). The caching hierarchy is constructed by routes between the origin site, caching routers, and users, such that less popular content is cached on *CR*s near to the origin site and more popular content is cached on *CR*s near to users. Therefore, it is difficult to evaluate the impact of TTL-based caching on network resources and performance because the characteristics in the distributed cache mechanism depend on the caching hierarchies connected by distributed cache nodes.

In this paper, we first propose an analytical model using matrix equations to evaluate the cache characteristics on multiplexed caching hierarchies of content and evaluate the validity of the proposed model and the impact of the TTL value.

III. Related Work

The modeling of efficient memory management is a major issue in content caching systems. Traditionally, there are content placement algorithms [2], [3] as a solution for *File Allocation Problems* [4], which minimize the cost imposed for content storage and queries, or maximize performance such as



(a) $TTL = 1$ (b) $TTL = 10$

Figure 3: Cache hit ratio when the requests per content with the rate λ input to a *CR* at an independent and identically distributed exponential interval and "total request rates of a content item [requests/sec]" and "TTL [sec]" to various values

distance to content. Baev *et al.* [2] propose a linear programming model which minimizes content placement cost and an approximation solution using a linear relaxation. Furthermore, Qui *et al.* [3] develop a method and compare it with some replica placement algorithms to solve a *K*-median problem for CDNs.

In contrast to the above-mentioned content placement problems, Borst *et al.* [5] formulate a liner programming model based on a hierarchical structure for content locations to minimize bandwidth costs through a distributed solution. In view of energy efficiency for content delivery networks, Guan *et al.* [6] build energy models of traffic transmission power and caching power for content delivery architectures such as "Conventional and decentralized server-based CDN", "Centralized server-based CDN using dynamic optical bypass", and CCN.

Furthermore, Carofiglio *et al.* [7] explore the impact of storage management on the cache performance per application in CCN and evaluate the effectiveness of static storage partitioning and dynamic management by priority-based weighted fair schemes combined with TTL-based caching. Moreover, they study the possibility of improving cache scalability in TTL-based caching without cache coordination.

However, these proposals don't discuss the cache characteristics using TTL of content and the impact of TTL on network resources and cache performance on the multiplexed caching hierarchies. Therefore in this paper, we construct a model to analyze the cache characteristics using TTL and evaluate the cache performance for the TTL value.

IV. Analytical Model

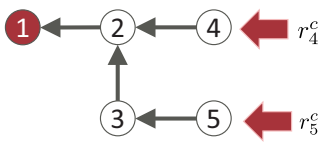
We first propose the evaluation model to analyze the cache performance using TTL of a content in the distributed cache system having multiplexed caching hierarchies.

In TTL-based caching, the cache probability of content c having request rates λ^c to a *CR* can be expressed by the following function [8].

$$\boldsymbol{f}(\lambda^c, TTL^c) = 1 - e^{-\lambda^c TTL^c}$$

As shown in Figure 3, we demonstrate that this statistical function can provide a good approximation of the cache hit

 ISBN: 978-1-61208-318-6



$$\mathbf{\Lambda}^c[s] = \begin{bmatrix} \lambda^c_{(1,4)}[s] & \lambda^c_{(1,5)}[s] \\ \lambda^c_{(2,4)}[s] & \lambda^c_{(2,5)}[s] \\ \lambda^c_{(3,4)}[s] & \lambda^c_{(3,5)}[s] \\ \lambda^c_{(4,4)}[s] & \lambda^c_{(4,5)}[s] \\ \lambda^c_{(5,4)}[s] & \lambda^c_{(5,5)}[s] \end{bmatrix} \quad \mathrm{R}^c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ r_4^c & 0 \\ 0 & r_5^c \end{bmatrix}$$

$$\mathrm{D}^c[s] = \begin{bmatrix} 0 & e^{-\sum_{j=4}^5 \lambda^c_{(2,j)} TTL^c} & 0 & 0 & 0 \\ 0 & 0 & e^{-\sum_{j=4}^5 \lambda^c_{(3,j)} TTL^c} & e^{-\sum_{j=4}^5 \lambda^c_{(4,j)} TTL^c} & 0 \\ 0 & 0 & 0 & 0 & e^{-\sum_{j=4}^5 \lambda^c_{(4,j)} TTL^c} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4: An example of matrices $\mathbf{\Lambda}^c$, $\mathbf{R}^c$, and D^c for the request propagation on the delivery tree having origin 1

ratio of content at a *CR* under the assumption that content requests arrive as a Homogeneous Poisson Process. We next show the matrix model of request propagation of content in TTL-based caching on caching hierarchy of content. The propagation of each request (*Interest*) of content c on its caching hierarchy is expressed by the following model.

$$\mathbf{\Lambda}^c[s+1] = \mathrm{D}^c[s] \cdot \mathbf{\Lambda}^c[s] + \mathbf{R}^c, \forall c \tag{1}$$

Under the condition that M, N, and s are the number of *CR*s, the number of sites having requesting users, and the number of steps that each request propagates to the next *CR*, respectively, we define $\mathbf{\Lambda}^c$ as the $M \times N$ matrix consisting of the request rates $\lambda^c_{(i,j)}$ of content c from the requesting user in site j to CR_i and $\mathbf{R^c}$ as the $M \times N$ matrix of which elements are the request rates r_i^c of content c from users in site i.

$$\mathbf{\Lambda}^c[s] := [\lambda^c_{(i,j)}]_{M \times N}$$

$$[\mathbf{R}^c]_{i,j} := \begin{cases} r_i^c, & \text{when } CR_i \text{ is located on } j\text{-th} \\ & \text{site having requesting users} \\ 0 & \text{otherwise} \end{cases}$$

D^c is the $M \times M$ matrix of request propagation for content c as follows.

$$[\mathrm{D}^c]_{m,n} := \begin{cases} 1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(n,k)}[s], TTL^c), \\ \quad m = parent\_node(n) \\ 0 \quad \text{otherwise} \end{cases} \tag{2}$$

Here we defined the condition that CR_m is a parent node of CR_n as "$m = parent\_node(n)$". In Figure 4, we show an example of these matrices for the delivery tree having origin 1.

In the iterative matrix equation, we can consider the request propagation process and data caching at each *CR* for content requested from each site. Moreover, the steady state of network resources and cache performance per content can be derived by iteratively calculating the equation s_{max}-times, which is the maximum number of hops from each site having requesting users to its origin site.

Using the proposed solution, we can model the system state and the cache performance for content c such as

- memory usage per content in each *CR*,
- the total amount of transmission data in the network,
- power consumption which is the sum of "cache allocation power" and "traffic transmission power",
- cache hit ratio per content which is the probability that the content is cached in the network, and
- average hop length per content.

A. Memory Usage

The memory usage of content c at CR_i is derived using the data size θ_c of content c as follows.

$$\mathrm{U}_i^c := \theta_c \boldsymbol{f}(\sum_k^N \lambda^c_{(i,k)}, TTL^c). \tag{3}$$

B. Transmission Data

The total amount of data delivery of content c through all *CR*s is derived as

$$\mathrm{Dt}^c := \theta_c \sum_j^N Tr_j^c \tag{4}$$

using the following vector consisting of the cumulative number of traffic flows Tr_j^c through each *CR* on the delivery route for content c having origin site o requested by users in site j.

$$\begin{aligned} \mathbf{Tr}^c &:= [Tr_1^c \cdots Tr_j^c \cdots Tr_N^c]^{\mathrm{T}} \\ &= (\mathbf{H} * \mathbf{\Lambda}^c)^{\mathrm{T}} \begin{bmatrix} \boldsymbol{f}(\sum_k^N \lambda^c_{(1,k)}, TTL^c) \\ \vdots \\ \boldsymbol{f}(\sum_k^N \lambda^c_{(M,k)}, TTL^c) \end{bmatrix} \\ &\quad + (\mathbf{H}[o,] * \mathbf{\Lambda}^c[o,])^{\mathrm{T}} (1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(o,k)}, TTL^c)) \end{aligned} \tag{5}$$

Here, we define "$*$" as the element-wise product of a matrix or vector and $\mathbf{H} = [h_{(i,j)}]_{M \times N}$ as the matrix consisting of shortest hop length $h_{(i,j)}$ from CR_i to CR_j.

Moreover, the second term in (5) presents the amount of transmission data which aren't cached on the network.

 ISBN: 978-1-61208-318-6

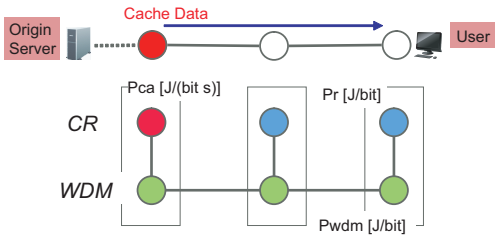


Figure 5: Network model

TABLE I: VARIABLES IN THE PROPOSED MODEL

Variable	Definition
M	The number of *CR*s
N	The number of sites having requesting users
θ_c	Data size of content c
$\lambda^c_{(i,j)}$	Request rates to CR_i for content c requested by users in site j
r^c_i	Request rate of content c requested by users in site i
TTL^c	TTL of content c at CR_i
U^c_i	Memory usage of content c at CR_i
Dt^c	Total amount of data delivery of content c through all *CR*s
Tr^c_j	Cumulative number of traffic flows through each *CR* on the delivery route for content c requested by users in site j
CHR^c	Cache hit ratio of content c in the network
AHL^c	Average hop length of content c
Hp^o	Hop length from origin server to the content router in origin site o
CP^c	Total power consumption [J] for data storage of content c in 1 sec
TP^c	Total power consumption [J] delivering content c on the delivery routes
P_{ca}	Power density for storage [J/(bit·s)]
P_r	Power density of a router [J/bit]
P_{wdm}	Power density of a WDM node [J/bit]

C. Power Consumption

We consider total power consumption based on *Energy Proportional Networks* [9], [10] in which power consumption of each device is proportional to its usage for a network composed of *CR*s and Wavelength Division Multiplexing (WDM) nodes in Figure 5. In this paper, we assume 1 sec as time unit.

Cache allocation power: CP^c [J] for storing data of content c in 1 sec, i.e., the total power consumed by storing content c on each *CR* in the network, is defined as

$$\mathrm{CP}^c := \theta_c P_{ca} \sum_i^M \boldsymbol{f}(\sum_k^N \lambda^c_{(i,k)}, TTL^c), \quad (6)$$

where P_{ca} is the memory power density [J/(bit· s)].

Traffic transmission power: TP^c [J] i.e., the total power consumed by network devices when data of content c are delivered on the shortest routes, is derived as

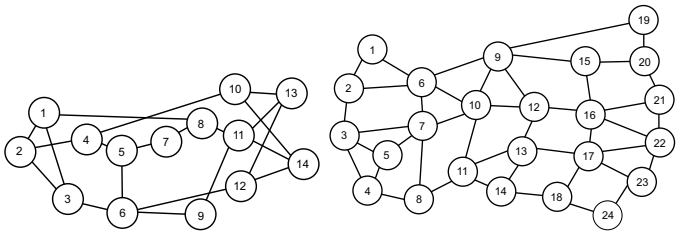
$$\mathrm{TP}^c := (P_r + P_{wdm})\mathrm{Dt}^c, \quad (7)$$

where P_r and P_{wdm} are the power densities [J/bit] of a router and of a WDM node along the delivery routes, respectively.

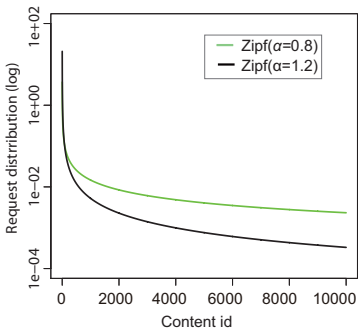
D. Cache Hit Ratio

The cache hit ratio of content c having origin o in the network is derived as

$$\mathrm{CHR}^c := 1 - \frac{\sum_j^N \lambda^c_{(o,j)} \left(1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(o,k)}, TTL^c)\right)}{\sum_j^N r^c_j}. \quad (8)$$



(a) Test topology A (b) Test topology B



(c) Request distribution r^c_i

Figure 6: Evaluation conditions

E. Average Hop Length

The average hop length of content c having origin o is derived as

$$\mathrm{AHL}^c := \frac{\sum_j^N \left(Tr^c_j + \mathrm{Hp}^o \lambda^c_{(o,j)} (1 - \boldsymbol{f}(\sum_k^N \lambda^c_{(o,k)}, TTL^c)\right)}{\sum_j^N r^c_j}. \quad (9)$$

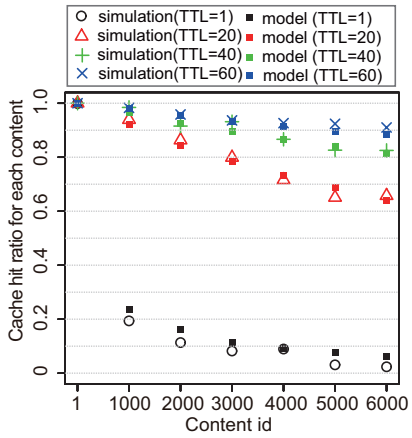
The second term of the numerator is a penalty for the hop length of content c, which isn't cached on any *CR*s in the network and for which a request reaches its origin server and Hp^o is the hop-length from the origin server to the content router in origin site o and is used as penalty if the content is not found in the network. All variables in the proposed model are summarized in Table I.

V. Evaluation using the Proposed Model

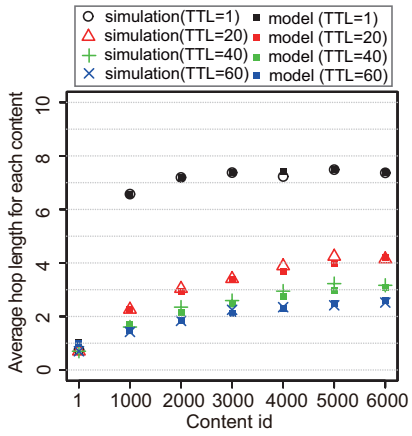
We evaluate the cache characteristics in TTL-based caching when changing the TTL value of content. The evaluation conditions are set to the following.

- **Test networks:** NSF topology with 14 *CR*s (Topology A), cf. Figure 6(a) / US-backbone topology with 24 *CR*s (Topology B), cf. Figure 6(b). The maximum number of hops (s_{max}) is 5 in Topology A and 7 in Topology B. Furthermore, we assume that the memory size of each *CR* is infinite and each site has requesting users for all content items, which means M is equal to N. For the evaluation, we set Hp^o to 5 as the penalty of hop length.

 ISBN: 978-1-61208-318-6

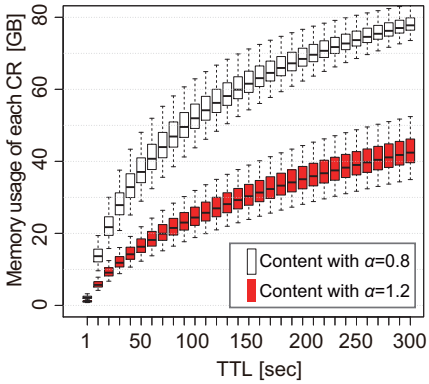


(a) Cache hit ratio

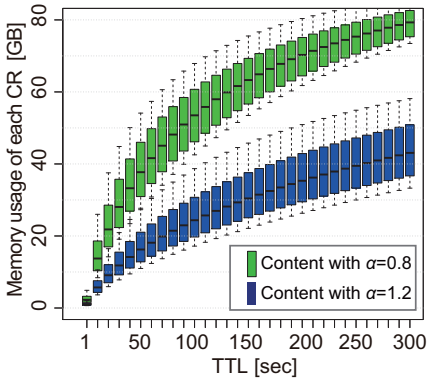


(b) Average hop length

Figure 7: Cache performance estimated by the proposed model and calculated by simulations for different content ids



(a) Topology A



(b) Topology B

Figure 8: Box plot of memory usage at each *CR* when the TTL value is changed

- **Content information:** Zipf-distributed requests from each site i for $K = 10000$ content items are defined as $r_i^c = \gamma k^{-\alpha}/c$, $c = \sum_{k=1}^{K} k^{-\alpha}$, cf. Figure 6(c). We set α to 0.8 for User Generated Content (UGC) and 1.2 for VoD [12] and γ to 100 [requests/sec]. Furthermore, the origin site t of content ID k is set randomly based on a uniform distribution. The content size is geometrically distributed with mean 10 MB [13].

A. *Verification of the Proposed Model*

To verify the proposed model, we compare the cache performance using the model with that measured by simulations for 7 content items with $\alpha = 0.8$ in Topology A. In the evaluations, we set the TTL value as $\{1, 20, 40, 60\}$ [sec].

Figure 7 shows that the cache hit ratio and average hop length for each content provide suitable approximations of the simulation results. As a result, we see that the proposed model can express the statistical characteristics for TTL-based caching.

B. *Impact of TTL*

For the next evaluation, we define TTL as the same value for each content which is changed from 1 [sec] to 300 [sec] on the assumption that the TTL value is signaled in the data header of content.

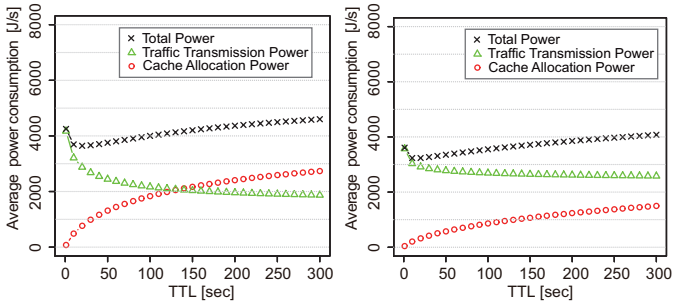
Figure 8 shows the memory usage at each *CR* when the TTL value is changed. In these results, the memory usage of each *CR* becomes larger as the TTL value becomes larger. Moreover, the memory usage for content with $\alpha = 0.8$ is larger than that for content with $\alpha = 1.2$ because less popular content with $\alpha = 0.8$ has higher request rates and is easier to be cached than that with $\alpha = 1.2$.

Furthermore, Figure 9 shows the power consumption of the target network according to the change of the TTL value using the power densities of network devices shown in Table II. In addition, Figure 10 presents cache hit ratio for all content items calculated by (8) and average hop length for all content items calculated by (9).

 ISBN: 978-1-61208-318-6

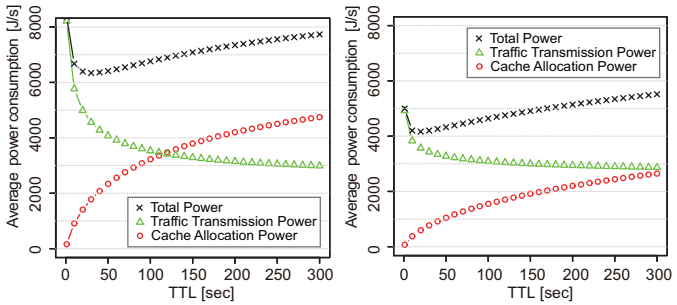
TABLE II: POWER DENSITY PARAMETERS

Device (Product)	Power / Spec	Power Density
DRAM	10 W / 4 GB	$P_{ca} = 3.125 \times 10^{-10}$ J/(bit · s)
Content Router (CRS-1)	4185 W / 320 Gbps	$P_r = 1.3 \times 10^{-8}$ J/bit
WDM (FLASHWAVE9500)	800 W / 480 Gbps	$P_{wdm} = 1.67 \times 10^{-9}$ J/bit



(a) Content with $\alpha = 0.8$ in Topology A

(b) Content with $\alpha = 1.2$ in Topology A



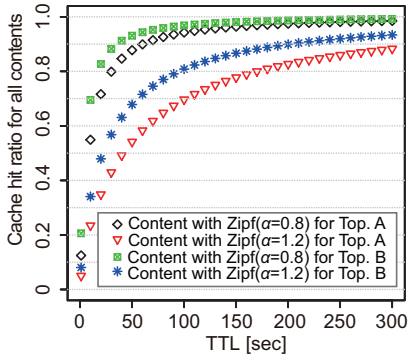
(c) Content with $\alpha = 0.8$ in Topology B

(d) Content with $\alpha = 1.2$ in Topology B

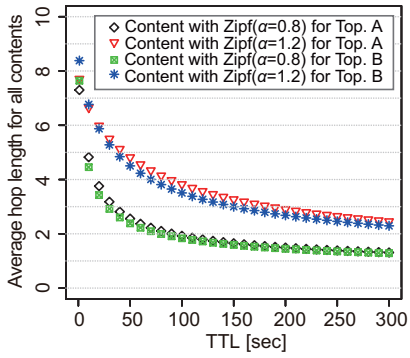
Figure 9: Power consumption of the network when the TTL value is changed

Figure 9 demonstrates the tradeoff between *cache allocation power* and *traffic transmission power* for the change of the TTL value. Figures. 9(a) and (c) show that there is a point reversing the relation of *cache allocation power* and *traffic transmission power* for the TTL value. Therefore, the energy impact of TTL is also different depending on the network conditions and the proposed model can search for the energy efficient TTL in consideration of the tradeoff of power consumption.

Meanwhile in Figure 10(a), the cache hit ratio is also low in the region of the TTL values leading to lower power consumption. Therefore, we should consider the relation between cache hit ratio and power consumption to search for the energy efficient TTL. Furthermore, Figure 10(b) shows that the average hop length of all content items becomes smaller as the TTL becomes larger. In these results, approaching the average hop length of 1 hop means that all content items are cached



(a) Cache hit ratio for all content items



(b) Average hop length for all content items

Figure 10: Cache performance when the TTL value is changed

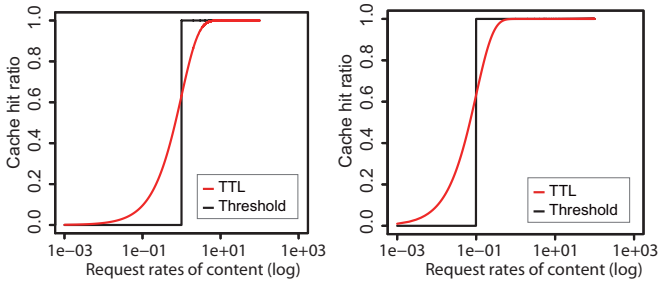
in all *CR*s. The cache hit ratio approaches to around 100 % as the average hop length is approaching to 1 and the memory usage becomes larger. Therefore, using the proposed model, we can analyze the cache characteristics in the distributed cache system and provide a design guideline for TTL of content in view of energy efficiency or efficient memory usage in each *CR*. Next, we introduce an energy efficient cache mechanism using TTL as application and demonstrate the effectiveness of the energy efficient TTL.

VI. Application to a Cache Mechanism using Energy Efficient TTLs

In consideration of energy efficiency in content dissemination networks, we previously proposed an ILP model to design the most energy efficient cache locations taking the multiplexed caching hierarchies into account [14].

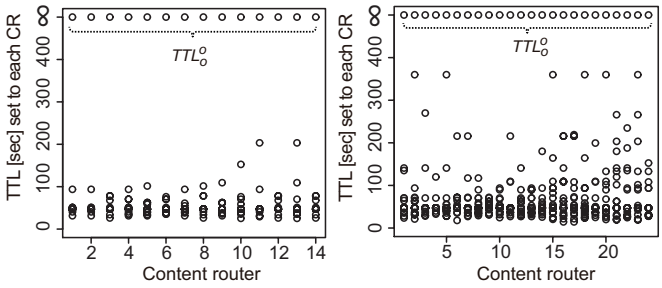
In [15], we proposed the threshold-based cache mechanism to locally search for locations which are near to the most energy efficient locations. In threshold-based caching, every *CR* automatically pre-designs a threshold of request rates of content using local information on each caching hierarchy before cache operation and the content data are cached when the request rate of the content is above a pre-designed threshold or isn't cached when the request rate is below that threshold.

 ISBN: 978-1-61208-318-6



(a) $TTL = 1$, $Th = 1$ (b) $TTL = 10$, $Th = 0.1$

Figure 11: Comparison of cache hit ratio at a *CR* with threshold-based caching and TTL-based caching



(a) Topology A (b) Topology B

Figure 12: Energy efficient TTL for each topology

In threshold-based caching, CR_i has the threshold Th_i^o [requests/sec] of request rates for origin site o of content, which is uniquely determined for each delivery tree in the target network. Furthermore, we can express the request propagation matrix D_{th}^c of threshold-based caching in the proposed model as

$$[\mathrm{D}_{th}^{c\in \mathrm{C_o}}]_{m,n} := \begin{cases} 1 & \forall m = parent\_node(n) \\ & \wedge \sum_k^N \lambda_{(n,k)}^c[s] < Th_n^o \\ 0 & \text{otherwise} \end{cases} . \quad (10)$$

$\mathrm{C_o}$ is the set of content items having origin o.

In this paper, we propose an approximation method using TTL of threshold-based caching because TTL-based caching can realize a more simple cache management by just updating the TTL counter of content without having to measure the request rates of content like in threshold-based caching.

We derive the approximation method using TTL of threshold-based caching as follows.

$$TTL_i^o = \frac{1}{Th_i^o}, \forall i, o \quad (11)$$

Here, TTL_i^o is set to CR_i and defined as a different value for each origin site o of content. In Figure 11, we evaluate the cache hit ratio of content at a *CR* for threshold-based caching and the energy efficient TTL-based caching. As a result, we see

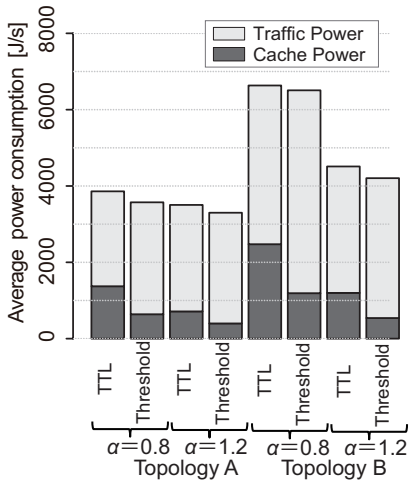


Figure 13: Power consumption for energy efficient TTL-based caching and threshold-based caching

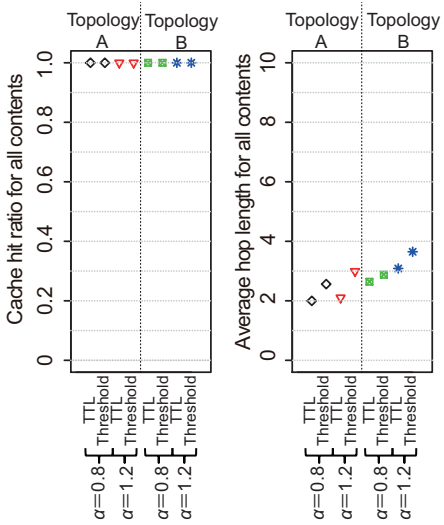


Figure 14: Cache performance for energy efficient TTL-based caching and threshold-based caching

that the cache hit ratio in the energy efficient TTL can provide similar characteristics to that in threshold-based caching.

Using (1), we can derive the request propagation matrix $\mathrm{D}_{ttl}^{c\in \mathrm{C_o}}$ using the energy efficient TTL as

$$[\mathrm{D}_{ttl}^{c\in \mathrm{C_o}}]_{m,n} := \begin{cases} 1 - \boldsymbol{f}(\sum_k^N \lambda_{(n,k)}^c[s], \frac{1}{Th_n^o}), \\ \qquad \forall m = parent\_node(n) \\ 0 \quad \text{otherwise} \end{cases} . \quad (12)$$

In these cache mechanisms, the threshold Th_o^o and the TTL $\frac{1}{Th_o^o}$ for content having origin o at CR_o are defined as 0 and ∞, respectively. Therefore, all content items are always cached in the network unless memory overflow occurs in each *CR*.

Here, we demonstrate the effectiveness of the energy efficient TTL based on the same conditions as in Section V. Figure 12 shows the TTL values derived by (11). In these

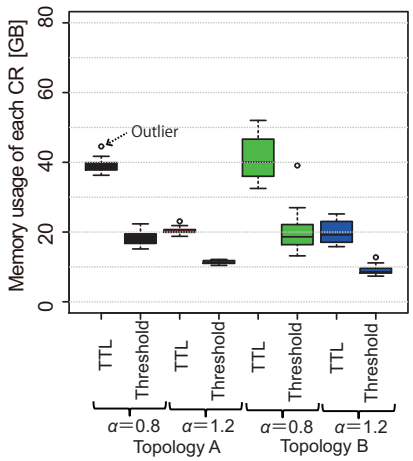


Figure 15: Box plot of memory usage for TTL-based caching and threshold-based caching

results, TTL_o^o at CR_o in origin site o is infinite and the other TTLs are derived as different values for each target network.

In Figure 13 and Figure 14, we compare the total power consumption and the cache performance for two mechanisms using the energy efficient TTLs and thresholds of request rates, respectively. In these results, the total power consumption in the energy efficient TTL-based caching is near to that in threshold-based caching. Moreover, the cache hit ratio is always 100% because TTL_o^o and Th_o^o $(\forall o)$ are infinite and 0. The average hop length in TTL-based caching is slightly smaller than that in threshold-based caching because the memory usage in TTL-based caching is larger than that in threshold-based caching as shown in Figure 15.

VII. Conclusion

We proposed an analytical model to evaluate the cache characteristics of a distributed cache system like CCN. The proposed model is expressed by iterative matrix equations and can evaluate the impact of TTL-based caching on network resources and cache performance on multiplexed caching hierarchies. In the evaluations, we verified the validity of cache characteristics estimated by the proposed model under the assumption that content requests are generated at an independent and identically distributed exponential interval and analyzed the impact on memory usage, power consumption, cache hit ratio, and average hop length when changing the TTL value of content. Furthermore, we introduced the energy efficient TTL to reduce the power consumption of the network and evaluated its effectiveness. Based on the proposed model, we showed that the energy efficient TTL-based caching can achieve a similar power consumption like threshold-based caching that searches for the most energy efficient cache locations and can realize shorter hop length than threshold-based caching.

As future work, we plan on enhancing the model in consideration of the limit of memory size and a different arrival process of content requests. Furthermore, we will study memory control mechanisms based on the theoretical model of TTL-based caching.

References

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content", in Proc. of CoNEXT'09, Rome, Italy, December, 2009, pp. 1–12.

[2] I. Baev, R. Rajaraman, and C. Swamy,"Approximation Algorithms for Data Placement in Arbitrary Networks", in Proc. of the 12th ACM-SIAM Symposium on Discrete Algorithms (SODA), Washington, DC, USA, January, 2001, pp. 661–670.

[3] L. Qiu, V. N. Padmanabhan, and G. M. Voelker, "On the Placement of Web Server Replicas", in Proc. of INFOCOM, Anchorage, AK, USA, April, 2001, pp. 1587–1596.

[4] Z. Drezner, "Facility Location: A Survey of Applications and Methods", Springer, Berlin, 1995.

[5] S. Borst, V. Gupta, and A. Walid, "Distributed Caching Algorithms for Content Distribution Networks", in Proc. of INFOCOM'10, San Diego, CA, USA, March, 2010, pp. 1–9.

[6] K. Guan, G. Atkinson, and D. C. Kilper, "On the Energy Efficiency of Content Delivery Architectures", in Proc. of the 4th IEEE International Conference on Communications (ICC) Workshop on Green Communications, Kyoto, Japan, June, 2011, pp. 1–6.

[7] G. Carofiglio, V. Gehlen, and D. Perino, "Experimental Evaluation of Memory Management in Content-Centric Networking", in Proc. of IEEE International Conference on Communications (ICC 2011), Kyoto, Japan, June, 2011, pp.1–6.

[8] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: modeling, design and experimental results", IEEE JSAC, 20(7), 2002, pp. 1305–1314.

[9] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework For Network Devices", in Proc. of NETWORKING'09, vol. 5550, Aachen, Germany, May, 2009, pp. 795–808.

[10] T. Harder, V. Hudlet, Y. Ou, and D. Schall, "Energy Efficiency is not Enough, Energy Proportionality is Needed!", in Proc. of DASFAA'11, Hong Kong, China, April, 2011, pp. 226-239.

[11] U. Lee, I. Rimac, D. C. Kilper, and V. Hilt, "Toward energy-efficient content dissemination", IEEE Network, vol. 25, no. 2, 2011, pp. 14–19.

[12] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, "Impact of traffic mix on caching performance in a content-centric network", in IEEE NOMEN'12, Workshop on Emerging Design Choices in Name-Oriented Networking, Orlando, Florida, USA, March, 2012, pp. 310–315.

[13] D. Rossi and G. Rossini, "Caching performance of content centric networks under multi-path routing", Technical Report, Telecom Paris Tech,. 2011.

[14] S. Imai, K. Leibnitz, and M. Murata, "Energy Efficient Content Locations for In-Network Cachingin Proc. of APCC'12, Jeju, Korea, October, 2012, pp. 554 – 559.

[15] S. Imai, K. Leibnitz, and M. Murata, "Energy-Aware Cache Management for Content-Centric Networking", in Proc. of First International Workshop on Energy-Aware Systems, Communications and Security, Barcelona, Spain, March, 2013, pp. 1623 – 1629.

 ISBN: 978-1-61208-318-6

A Simplified Queueing Model to Analyze Cooperative Communication with Network Coding

José Marcos C. Brito
National Institute of Telecommunication - Inatel
Santa Rita do Sapucaí, Brazil
brito@inatel.br

***Abstract*—Cooperative communication and network coding are two important techniques to improve the performance of telecommunications networks. Chieochan and Hossain proposed an interesting algorithm using these techniques, called BE-ONC (Buffer Equalized Opportunistic Network Coding), and analyzed its performance via simulation. In this paper, we proposed a simplified analytical queuing model to investigate the performance of wireless networks with cooperative communication and network coding. The proposed model was implemented and used to evaluate the performance of the BE-ONC algorithm. We also compared the performance of a system with cooperative communication and network coding to the performance of a system without cooperation.**

Keywords- cooperative communication; network coding; queueing mode; performance

I. INTRODUCTION

The traffic in telecommunications networks has grown exponentially. This is a consequence of the growth of the Internet, the development of new multimedia applications, and the huge proliferation of mobile terminals. To transmit this enormous traffic with QoS (Quality of Service), it is necessary to improve the performance of current telecommunications networks. One approach that has been widely studied as a solution to improve the performance of telecommunications networks is to use cooperative networks or cooperative communication [1-6].

Cooperation can be defined as the process of working together, as opposed to working separately (in competition) [1]. The basic idea of cooperative communication is to establish an additional path, via a relay node, connecting the source node to the destination node [1, 5]. Some cooperation techniques proposed in the literature are disclosed in [2,5,7], and those methods are classified as follows:

- Amplify-and-Forward (AF) - in this case, the relay amplifies the signal received from the source node and transmits this signal to the destination node.
- Decode-and-Forward (DF) - in this technique, the relay decodes the packet and re-encodes it prior to forwarding the packet to the destination node.
- Coded Cooperation - this is a technique that integrates cooperation into channel coding. The basic idea is that each user attempts to transmit incremental redundancy to its partner [5].
- Cooperative ARQ (Automatic Repeat reQuest) Protocols [7] – in this technique, the source node broadcasts its packets to the destination and relay nodes. If the destination node correctly receives the packet, the transmission is complete. However, if the packet is received incorrectly in the destination node but is correctly received in the relay node, re-transmission of the packet is performed by the relay. Finally, if packet errors are detected by the destination and relay nodes, the source node re-transmits the packet.

Another way to establish a cooperative communication is to use the technique called network coding. The theory of network coding was introduced by Ahlswede et al. [8]. In this paper, we are interested in a cooperative communication technique using network coding called BE-ONC (buffer equalized opportunistic network coding), which was proposed by Chieochan and Hossain [9] for Wi-Fi networks. In that work, the performance of their algorithm has been analyzed using simulations only.

The first goal of this paper is to propose a simplified queuing model to analyze the performance of the BE-ONC algorithm and subsequently compare the performance of a system with cooperation and network coding to that of a system without cooperation. The delay required for the successful transmission of a packet in the network is used as the parameter to assess system performance.

The remainder of this paper is organized as follows: in Section II, we summarize the BE-ONC algorithm and present the queuing model used in [9] to simulate the performance of this algorithm. In Section III, we present a simplified analytical model to evaluate the performance of the BE-ONC algorithm and reveal some of the numerical results. The conclusions and a preview of future initiatives are presented in Section IV.

II. THE BE-ONC ALGORITHM [9]

This section summarizes the BE-ONC algorithm proposed in [9] as a cooperative communication algorithm based on network coding for Wi-Fi networks.

The network analyzed in [9] is composed of two wireless users, one relay node, and one access point (AP) as

 ISBN: 978-1-61208-318-6

illustrated in Figure 1. The wireless users broadcast packets to the relay and to the AP. After transmitting a packet, a wireless user waits for a positive acknowledgment (ACK) from the relay or the AP. If an ACK is received, the packet is deemed to have been successfully transmitted and is removed from the user's queue. If an ACK is not received, the wireless user re-transmits the packet.

The relay receives packets from wireless user 1 and wireless user 2. If a packet is correctly received by the relay but not by the AP, it is queued in the relay to be re-transmitted to the AP. The relay tries to combine two packets using an XOR (eXclusive OR) operation before transmitting them to the AP. Although the relay is allowed to re-transmit packets, it does not generate traffic [9].

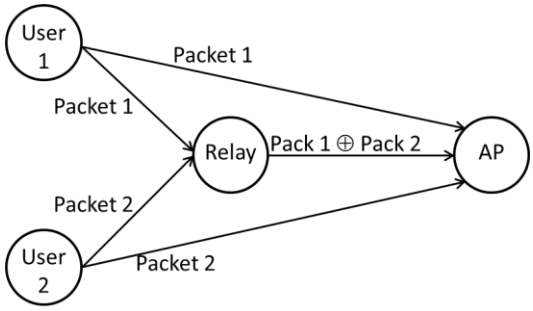


Figure 1. Wireless cooperative relay network with network coding.

Packets transmitted by wireless users are classified as non-urgent and urgent. The relay maintains two buffers, and only packets not received (or received with error) by the AP are queued in the relay node. If both buffers are empty, an incoming packet from user *j* (*j* = 1 or 2) is queued on queue *j*. Non-urgent packets coming from user *j* (*j* = 1 or 2) are also queued on queue *j*. Urgent packets are queued in the less congested buffer.

If the relay has packets in both queues (1 and 2), it combines the head-of-line (HOL) packets (using an XOR operation) and transmits the resulting packet to the AP. If the combined packet is correctly received by the AP, both HOL packets are removed from the relay's queue. If only one queue has a packet, the relay transmits the HOL packet of this queue. Again, if the packet is correctly received by the AP, it is removed from the relay's queue.

Figure 2 illustrates the queuing model presented in [9] that was used to analyze the performance of the algorithm. Again, the analyses performed in [9] are executed via simulations only.

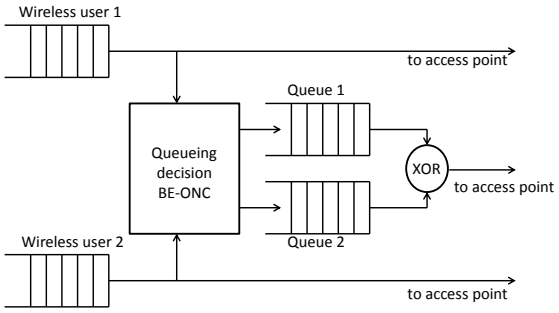


Figure 2. Queueing model presented in [9].

III. THE PROPOSED SIMPLIFIED QUEUEING MODEL

The mathematical analysis of the queuing model used to accomplish the simulations performed in [9] is quite difficult. To overcome this problem, we propose a simplified queuing model of the cooperative network in this paper.

Following [9], we assume that packets arrive randomly at each wireless user's buffer according to a Poisson process.

In addition, to model the relay's queue as a Markovian process, we consider the service time to be exponentially distributed in all queues in the network.

The above assumptions are important to compute the packet delays using the relevant theoretical results pertaining to networks of queues presented in the literature.

Figure 3 illustrates the simplified queuing model proposed in this paper. In that figure, P_1 represents the probability of a packet being queued in the relay and is computed by:

$$P_1 = PER_{uap} \cdot \left(1 - PER_{ur}\right) \quad (1)$$

where PER_{uap} is the packet error rate in the wireless link between a wireless user and the AP, and PER_{ur} is the packet error rate in the wireless link connecting a wireless user and the relay.

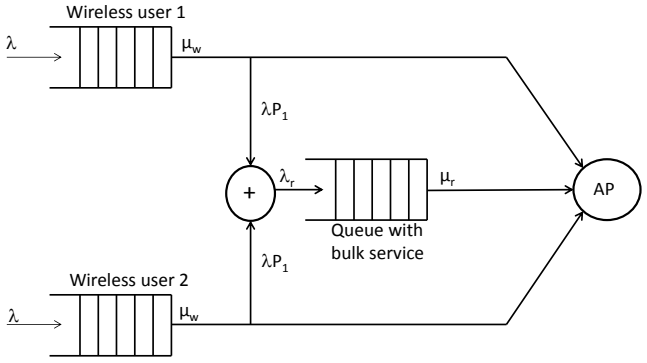


Figure 3. The proposed simplified queueing model.

The queue for each wireless user is modeled as an M/M/1 queue. The arrival rate in each user's queue is λ packets/second. Because the packet stays in the HOL of the

user's queue until it has been correctly received by the relay or the AP, the packet error rate in the wireless links (between a user and relay and between a user and AP) must be considered to calculate the real service time in the user's queue. Defining μ, in packets/second, as the capacity of the output link of a user's queue, the effective mean service time in the user's queue can be computed from:

$$E(t_{sw}) = \frac{1}{\mu_w} = \frac{1}{\mu} \cdot P \cdot \sum_{k=1}^{\infty} k(1-P)^{k-1} = \frac{1}{\mu \cdot P} \quad (2)$$

where P is the probability of a packet being received without error by the relay or AP. This probability can be written as a function of the packet error rate in the wireless links as:

$$P = 1 - (PER_{ur} \cdot PER_{uap}) \quad (3)$$

Substituting (3) in (2), we can rewrite the mean service time in the user's queue as:

$$E(t_{sw}) = \frac{1}{\mu_w} = \frac{1}{\mu \cdot [1 - (PER_{ur} \cdot PER_{uap})]} \quad (4)$$

Considering the M/M/1 model, the total time spent by a packet in a user's queue is determined as:

$$E(T_u) = \frac{1}{\mu_w - \lambda} \quad (5)$$

To account for the XOR operation on the HOL packets shown in Figure 2, the queue in the relay node is modeled as one with bulk service. The corresponding state transition diagram is illustrated in Figure 4. If only one packet is in the queue, it is immediately transmitted by relay. If two or more packets are in the queue, the relay executes an XOR operation between the two packets in the HOL and transmits the combined packet.

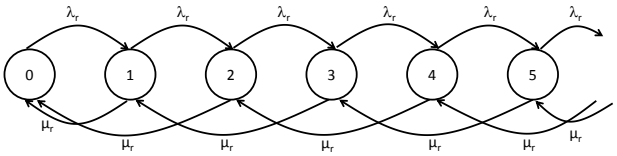


Figure 4. State transition diagram of the relay's queue.

From Figure 3, the arrival rate in the relay's queue is given by:

$$\lambda_r = 2\lambda \cdot P_1 \quad (6)$$

To compute the effective mean service time in the relay's queue, we need to consider the packet error rate in the link between the relay and the AP. Defining μ as the capacity of the output link in the relay node, the mean service time in the relay's queue can be computed from:

$$E(t_{sr}) = \frac{1}{\mu_r} = \frac{1}{\mu \cdot [1 - PER_{rap}]} \quad (7)$$

where PER_{rap} is the packet error rate in the wireless link between the relay and the AP.

The total time spent in the queue with bulk service (illustrated in Figure 4) can be computed by [10] [11]:

$$E(T_r) = \frac{r_0}{\lambda_r(1-r_0)} + \frac{1}{\mu_r} \quad (8)$$

where r_0 is the positive root of the operator equation (9) having a value less than 1 [10]:

$$\mu \cdot r^3 - \mu \cdot r - \lambda_r \cdot r + \lambda_r = 0 \quad (9)$$

Solving (9), the only positive root with a value less than 1 is given by:

$$r_0 = \frac{-\mu + \sqrt{\mu^2 + 4\mu\lambda}}{2\mu} \quad (10)$$

Finally, we can compute the mean time required to transmit a packet (without error) from the wireless user to the AP as:

$$E(T_t) = E(T_u) \cdot (1 - P_1) + [E(T_u) + E(T_r)] \cdot P_1 \quad (11)$$

To compare the performances of systems with different capacities, it is advantageous to normalize equation (11) as a function of a packet's transmission time (1/μ), resulting in the normalized delay:

$$E(T_{tn}) = \{E(T_u) \cdot (1 - P_1) + [E(T_u) + E(T_r)] \cdot P_1\} \cdot \mu \quad (12)$$

Figure 5 illustrates the behavior of the normalized delay as a function of the utilization factor in the wireless link ρ. The utilization factor is the ratio of the load for each wireless user λ relative to the transmission capacity of its output wireless link μ.

The packet error rate in a wireless link is strongly dependent of the link quality. For example, packet error rates from 0.018 to 0.738 are reported in [12]. The results presented in Figure 5 consider PER_{uap} = 0.3 and PER_{ur} = PER_{rap} = 0.1.

Finally, it is interesting to compare the normalized delay in the cooperative system with network coding to the performance of a system without cooperation (without the relay node). To be fair in this comparison, the transmission capacity of the relay is equally divided between the two wireless users. Thus, the transmission capacity for each wireless user is 1.5μ. In this case, the total packet-transmission delay from a wireless user to the AP is computed from:

 ISBN: 978-1-61208-318-6

$$E(T_{twc}) = \frac{1}{1.5\mu \cdot (1 - PER_{uap}) - \lambda} \quad (13)$$

and the normalized delay without cooperation is given by:

$$E(T_{twcn}) = \frac{\mu}{1.5\mu \cdot (1 - PER_{uap}) - \lambda} \quad (14)$$

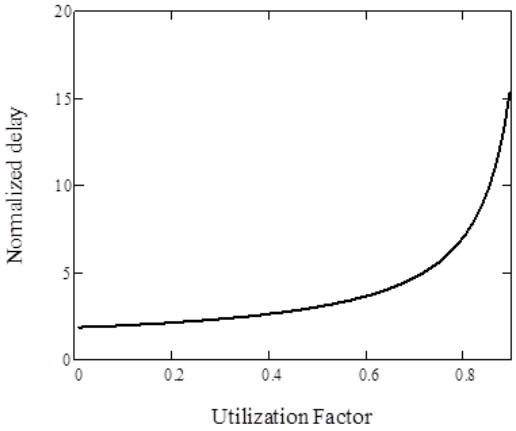


Figure 5. Normalized delay as a function of utilization factor, ρ, considering PER_{uap} = 0.3 and PER_{ur} = PER_{rap} = 0.1.

Figures 6 and 7 compare the performance of the system with cooperation and network coding vis-à-vis the system without cooperation. In Figure 6, we consider PER_{uap} = 0.3 and PER_{ur} = PER_{rap} = 0.1, and in Figure 7, PER_{uap} = 0.4 and PER_{ur} = PER_{rap} = 0.1.

We can observe that the system with cooperation performs better than the system without cooperation once a given packet error rate threshold in the link between the wireless user and the AP has been exceeded. The threshold is a function of the following parameters: PER_{uap}, PER_{ur}, PER_{rap} and ρ. To investigate the value of this threshold, we define a performance factor, Δ, as the ratio between Equation 12 and Equation 14. The behavior of this parameter is illustrated in Figure 8 and Figure 9.

If the performance factor, Δ, is greater than 1, the system without cooperation performs better than the system with cooperation. If Δ < 1, the system with cooperation performs better than the system without cooperation.

Figure 8 shows the influence of the utilization factor, ρ, considering PER_{uap} = 0.4 and PER_{ur} = PER_{rap} = 0.1.

Figure 9 shows the influence of the packet error rate in the link between the wireless user and AP, PER_{uap}, considering ρ = 0.8 and PER_{ur} = PER_{rap} = 0.1.

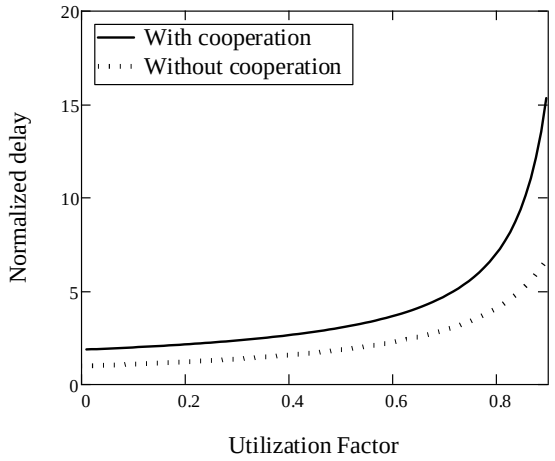


Figure 6. Comparing the normalized delay between systems with and without cooperation, considering PER_{uap} = 0.3 and PER_{ur} = PER_{rap} = 0.1.

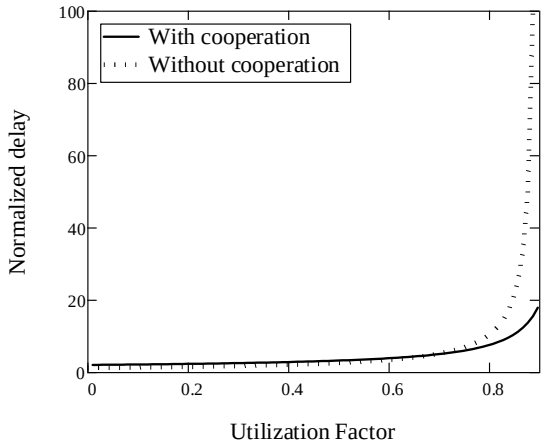


Figure 7. Comparing the normalized delay between systems with and without cooperation, considering PER_{uap} = 0.4 and PER_{ur} = PER_{rap} = 0.1.

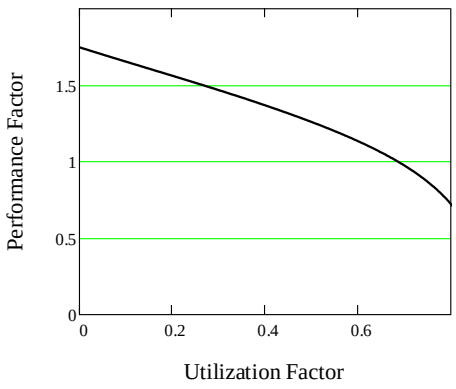


Figure 8. Performance Factor as a function of Utilization Factor, considering PER_{uap} = 0.4 and PER_{ur} = PER_{rap} = 0.1.

 ISBN: 978-1-61208-318-6

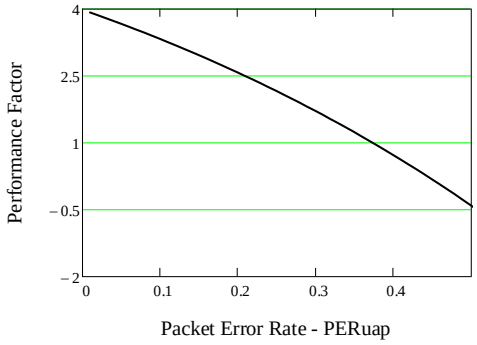


Figure 9. Performance Factor as a function of the Packet Error Rate in the link between the wireless user and AP, PER_{uap}, considering $\rho = 0.8$ and $PER_{ur} = PER_{rap} = 0.1$.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an analytical approach based on a simplified queuing model to analyze the performance of cooperative communication with network coding, and we used this model to evaluate an algorithm previously proposed in the literature. The parameter used to evaluate the performance characteristics is the delay to transmit a correct packet from a wireless user to an AP.

Additionally, we compare the performance of a system with cooperation and network coding to a system without cooperation. We concluded, concurrent with the literature, that cooperation increases the performance if the packet error rate in the direct link between the wireless user and AP is greater than a given threshold.

The main advantage of the proposed queuing model is its simplicity, making it easier to investigate the influence of system parameters on a network's performance. This type of model is very useful in that it provides valuable insight relative to the performance of the network.

The weakness of the proposed model is that the classification of the traffic (i.e., urgent and non-urgent traffic) used in the algorithm proposed in [9] is not considered in our model. In future endeavors, we intend to expand the model by incorporating the traffic classification aspects of the algorithm proposed in [9].

REFERENCES

[1] W. Zhuang and M. Ismail, "Cooperation in Wireless Communication Networks", IEEE Wireless Communications, April 2012, pp. 10-20.

[2] Y. Hong, W. Huang, F. Chiu and C. C. J. Kuo, "Cooperative Communications in Resource-Constrained Wireless Networks", IEEE Signal Processing Magazine, May 2007, pp. 47-57.

[3] Q. C. Li, R. Q. Hu, Y. Qian and G. Wu, "Cooperative Communications for Wireless Networks: Techniques and Applications in LTE-Advanced Systems, IEEE Wireless Communications, April 2012, pp. 22-29.

[4] C. Hoymann, W. Chen, J. Montojo, A. Golitschek, C. Koutsimanis and X. Shen, "Relaying Operation in 3GPP LTE: Challenges and Solutions, IEEE Communications Magazine, February 2012, pp. 156-162.

[5] A. Nosratinia, T. E. Hunter and A. Hedayat, "Cooperative Communications in Wireless Networks", IEEE Communications Magazine, October 2004, pp. 74-80.

[6] X. Tao, X. Xu and Q. Cui, "An Overview of Cooperative Communications", IEEE Communications Magazine, June 2012, pp. 65-71.

[7] J. Morillo-Pozo, D. Fusté-Vilella and J. García-Vidal, " Cooperative ARQ Protocols", Chapter 12 in Cooperative Wireless Communications, Edited by Y. Zhang, H. Chen and M. Guizani, Auerbach Publications, first edition, March 2009, pp.259-281.

[8] R. Ahsdwede, N. Cai, S. R. Li and R. W. Yeung, "Network Information Flow", IEEE Transactions on Information Theory, Vol. 46, No 4, July 2000, pp. 1204-1216.

[9] S. Chieochan and E. Hossain, "Cooperative Relaying in Wi-Fi Networks with Network Coding", IEEE Wireless Communications, April 2012, pp. 57-65.

[10] D. Gross and C. M. Harris, Fundamentals of Queueing Theory – second edition, John Wiley & Sons, 1985.

[11] L. Kleinrock, Queueing Systems Volume 1: Theory, John Wiley & Sons, 1975.

[12] B. Han and S. Lee, "Efficient Packet Error Rate Estimation in Wireless Networks", Proceedings of the 3rd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, TridentCom 2007, May 2007, pp. 21-23.

 ISBN: 978-1-61208-318-6

Improving Recovery in GMPLS-based WSON Through Crank-back Re-routing

Edgard Jamhour, Manoel Camillo Penna
Graduate School of Computer Science
Pontifícia Universidade Católica do Paraná - PUCPR
Rua Imaculada Conceição, 1155, 80215-901, Curitiba, Brasil
e-mail: jamhour@ppgia.pucpr.br, penna@ppgia.pucpr.br

***Abstract*—This paper defines and evaluates an unplanned technique based on Generalized Multi-Protocol Label Switching (re-routing) to restore lightpaths interrupted by failures in wavelength switched optical network. Compared to the pre-planned recovery techniques, the re-routing approach may significantly save network resources, but may suffer from longer recovery times and even fail to recover lightpaths, due to slow convergence of the information transported by Interior Gateway Protocol. To address this issue, we have used the crank-back extensions proposed by IETF, combined with a make-before-break strategy that re-uses resources from the broken lightpath to setup a recovery lightpath. We present an evaluation that permits to conclude about the performance of the proposed approach.**

Keywords-crank-bak re-routingt; make-before-brake; routing and wavelength assignment.

I. INTRODUCTION

Generalized Multi-Protocol Label Switching (GMPLS) defines a set of standards for managing lightpaths in Wavelength Switched Optical Network (WSON). The GMPLS control plane supports the following recovery techniques: protection, restoration, and re-routing [6]. In protection, recovery paths are planned and cross-connected before a failure occurs. It provides fast recovery times, but are costly because backup resources cannot be shared. In restoration, recovery paths are planned and resources are reserved in advance, but recovery paths are cross-connected only when a failure occurs. Restoration is less expensive than protection because multiple recovery paths may share the same wavelengths. However, it is still expensive because shared resources cannot be used by service paths even when there are no failures in the network. Re-routing refers to the unplanned recovery technique, where all the process of defining a recovery path and reserving resources in made after a failure occurs. Compared to the pre-planned recovery techniques, it may significantly save network resources, but may suffer from longer recovery times and fail to recover lightpaths, due to slow convergence of network information transported by Interior Gateway Protocol (IGP).

In a high capacity network, a single failure can interrupt a multitude of lightpaths and trigger a strong competition for resources. The network view will be outdated, and the setup of lightpaths planned with incorrect information will probably fail. To address this issue, we have used the crank-back extensions proposed by IETF [2], which define a flexible way to include additional information in the messages exchanged by the signaling protocol, i.e., Resource Reservation Protocol with Traffic Engineering (RSVP-TE). The extensions permit to include the information required to plan an alternate route in case of failure, and to modify the flow of the signaling messages to contour the parts of the network that are interrupted. Our recovery approach includes the following ideas. First, the nodes adjacent to the failure use the crank-back extensions to inform the ingress nodes about the information required to recover the interrupted lightpaths. Second, the recovery lightpath is planned by the ingress node using a load balance heuristic, which avoids the creation of bottlenecks and favors the reuse of resources. Third, recovery is performed using a Make-Before-Break (MBB) strategy, to reuse as much as possible the resources and cross-connects of the original lightpath that survive the failure. Finally, signaling is performed using a flexible segment re-rerouting strategy, permitting any node along the path to fix the information planned by the ingress node. MBB is pointed as being advantageous to improve the likelihood of a successful recovery (see [6], for example), but no previous work has detailed how it could be implemented in WSON.

The remaining of this paper is organized as follows. In Section 2, we review the WSON literature by focusing on improvements to RSVP-TE and crank-back. Section 3 explains the problems that may rise in an unplanned attempt to recover lightpaths and how we address the pointed issues. Section 4 presents the algorithms that compose our solution. The evaluation of the proposed method is found in Section 5. Finally, Section 6 presents the summary of our most important results and our vision about future research topics related to the subject.

II. RELATED WORK

Some improvements to RSVP-TE have been proposed to increase the likelihood of a successful label suggestion assignment during path creation. Sambo et al. [12] review several strategies that employ the label preference approach. The suggested vector object is introduced by Andriolli et al. [1] for networks with wavelength conversion capability. It collects information about the number of conversions that will be performed by intermediate nodes. This information permits the destination node to select a wavelength from the Label Set that minimizes the number of conversions. The suggested vector approach is further explored by Giorgetti et al. [3] to avoid contention of wavelengths due to outdated information in nodes that receive Path or Resv messages. The proposals previously mentioned improve RSVP-TE but

 ISBN: 978-1-61208-318-6

are not comparable to our work because they don´t make use of crank-back.

Planning protected lightpaths using a shared protection scheme is discussed by Munoz et al. [10]. The proposed extension of RSVP-TE includes information indicating which wavelengths in a Label Set are already being used by a protection lightpath. The same extensions are employed by Manolova ett al. [9], but considering networks with a limited number of wavelength converters. The paper extends the previous proposal by combining the idea of the suggested vector to reduce the number of wavelength converters along the path. Manolova et al. [7] [8] extend the same approach to include the sharing of optical regenerators. Giorgetti et al. [4] explore the use of suggested vector to improve Resv blocking, and evaluates the strategy in scenarios with or without crank-back attempts. The proposals discussed in this paragraph don't cover segment-based rerouting because the error messages always propagates to the ingress node, which is responsible for generating a new setup attempt.

More recently, some alternative approaches have been proposed. Pavani and Waldman [11] present a Routing and Wavelength Assignment (RWA) strategy with crank-back support based on Ant Colony Optimization (ACO) algorithm. The proposed strategy can be classified as segment-based re-routing, however, instead of using the RSVP-TE extensions, or IGP updates to propagate the crank-back information, the authors assume an ACO based algorithm that updates local state information of different aspects of the routing process. Chen et al. [13] propose a new routing protocol based on the concept of intensity gradient from an information source. It is based on a distance-vector routing scheme that enables the re-routing capability on every intermediate node, which maintains all possible link-disjoint routes to the destination node. The proposal includes a new signaling protocol that implements the information-diffusion-based routing. Because the proposals discussed in this paragraph are based in proprietary protocols, their corresponding approaches require a complete modification of the IGP algorithm and routing information presently used in GMPLS.

To the extent of our knowledge, the literature about the use of crank-back extensions in WSON is still weakly explored. There is nothing in the literature comparable to the study presented in this paper, in terms of exploring the signaling protocol extensions to define a method to recovery of lightpaths using a purely distributed approach that is robust against the problems caused by the slow convergence of IGP information. In a previous study, Jamhour and Penna evaluated [14] eight different network topologies to determine which network features favor the crank-back strategy, considering several network metrics, including some used in Social Network Analysis (SNA), allowing to find the criteria that permits to identify the situations in which the crank-back approach, or other re-route strategy is advantageous. However, the algorithms presented in this paper are totally new.

In special, we define an approach to coordinate the recovery attempts according two strategies: MBB and Break-Before-Make (BBM). In the next section, we present some examples of recovery problems caused by the IGP slow convergence and define the strategies to improve the likelihood of a successful restoration using a combined crank-back and re-routing strategy.

III. PROBLEM FORMULATION

In this section, we show how the MBB approach may be useful in WSON and why it may result in a temporary deadlock in some situations. We propose the use Notify messages to improve the ingress node perception about the possibility of completing a successful MBB recovery. The discussion in this section is based on the scenario in Figure 1. All links are supposed to have only two wavelengths at each direction. There are two uni-directional lightpaths created between the nodes 2 and 6 (represented by the square and circle symbols), and one uni-directional lightpath created between the node 1 and 6 (represented by the triangular symbols). The symbols in the links between nodes represent the direction and the wavelengths used by each lightpath. In this setup, no wavelength converter is used because lightpaths use the same wavelength in all links.

Suppose that link 4-5 fails. The failure is perceived by the adjacent nodes, and interrupts the three lightpaths. A Notify can be used to inform the ingress node of every lightpath affected by a failure (see Figure 2). It is not necessary that both nodes generate a Notify message about the same interrupted lightpath, but according to our recovery method, both nodes send the message.

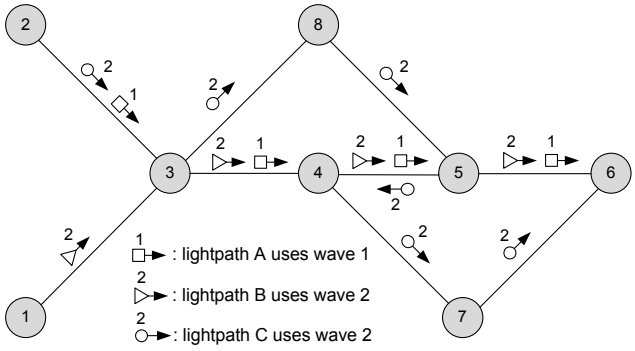


Figure 1. Sample scenario with three lightpaths.

In standard re-routing recovery, the ingress node must setup a recovery lightpath for each broken lightpath reported in the Notify message. The lightpath can be planned by the ingress node, or constructed in a distributed way. Each node in the network has its own view of the availability of resources. Ideally, a node should know about each wavelength available at each link, and the availability of wavelength converters in the nodes. However, flooding information about individual wavelengths is not practical. Moreover, in case of failure, this information is supposed to change very fast because several attempts of lightpath setups will be performed simultaneously. We assume that the only information available is the link state and the number of free wavelengths in each link.

 ISBN: 978-1-61208-318-6

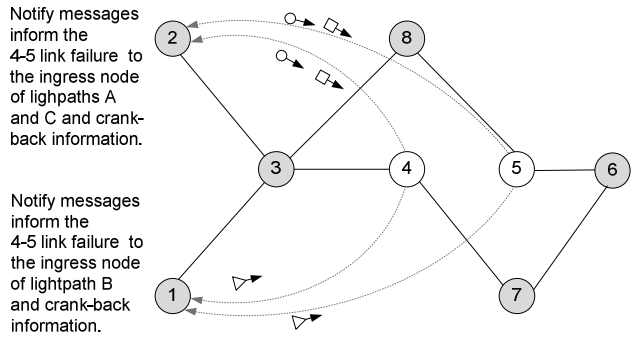


Figure 2. A Notify message is used to report the ingress node of each interrupted lightpath.

Figure 3 shows how the node 2 perceives the recovery options for the lightpath A, in a scenario without wavelength converters. The straight lines in the graph represent the status of the links based on IGP information, and the curved lines additional crank-back information supplied by the Notify messages. Without considering the availability of the wavelengths used by the original path, the recovery is unfeasible. The ingress node cannot perform an immediate recovery attempt without trying to reuse the wavelengths of the broken lightpath. The IGP information indicates that there is one wavelength available at the links 3-8 and 8-5. However, it is not possible to know if the wavelength "1" is available at these links. We use the Notify messages to indicate if the wavelength used by the broken lightpath can be successfully cross-connected to its adjacent edges. The cross-connect is possible if the same wavelength is available, or if it can be converted to an available wavelength. Node 4 informs the ingress node that a recovery attempt using MBB is possible for the link 4-7 and node 5 informs the same for link 8-5. The ingress node, however, does not have additional information about the links 3-8 and 7-6.

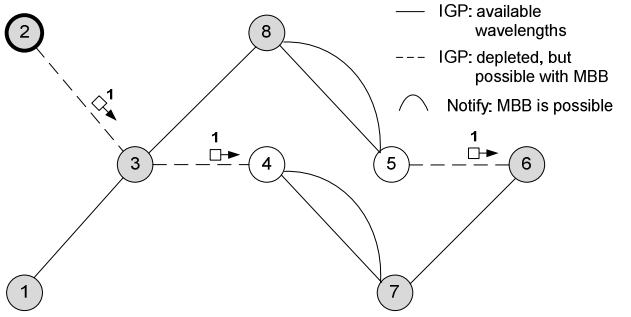


Figure 3. Recovery options for lightpath A perceived by the node 2.

The recovery options for lightpath B are computed by ingress node 1 (see Figure 4). Again, the node has no recovery options for lightpath B without considering the reuse of the wavelengths of the broken lightpath. If the decision is based exclusively on IGP information, node 1 will consider that MBB can be successful. However, the Notify messages sent by nodes 4 and 5 inform that it is not possible to cross-connect the wavelengths of the original lightpath to the links 4-7 and 8-5, because the wavelength is already in use, and the nodes have no wavelength converters. At present situation, the node has no immediate recovery option. Observe in the legend of the figure that we used the term “MBB is unlikely”, instead of “MBB is impossible”. MBB would be impossible if the wavelength required to perform cross-connect belongs to a lightpath that is not interrupted by the failure.

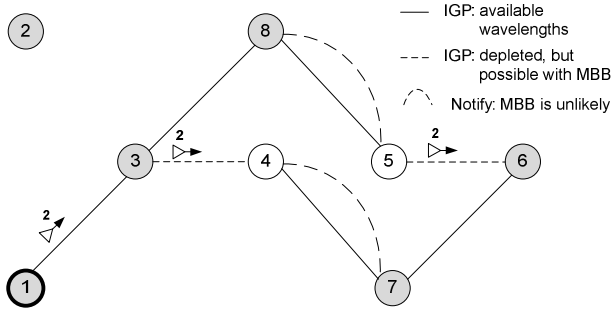


Figure 4. A Notify message is used to report the ingress node of each interrupted lightpath.

As indicated in Figure 5, if node 2 viewpoint of network resources is based solely on the information received by IGP, the recovery of lightpath C will be unfeasible (if the depleted links are removed, nodes 2 and 6 become disconnected). However, the Notify messages will indicate that a recovery may be possible in a near future, because some wavelengths are required to complete a MBB setup belongs to broken lightpaths. The situations of the lightpath B and C are similar, because none of them have an immediate recovery option.

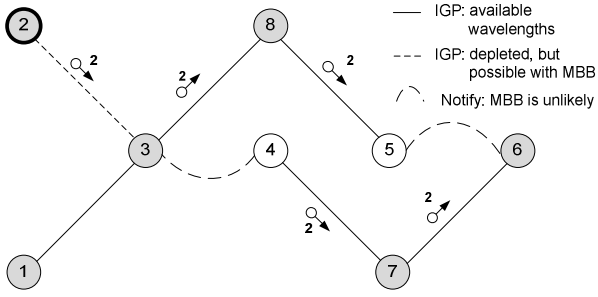


Figure 5. Recovery options for lightpath C perceived by the node 2.

The dynamic planning of recovery lightpaths may be done according to MBB or BBM. In the first, the original lightpath is teared down before the setup of the recovery lightpath. In the second, the resources are released only after the setup of the recovery lightpath is confirmed or aborted. If nodes 1 and 2 try MBB recovery, they would be in deadlock state (until the reservation is broken by the soft-state). On the other hand, if they try BBM recovery, it would be possible to

 ISBN: 978-1-61208-318-6

recover all lightpaths. However, BBM approach cannot be performed immediately, because the release of the resources of the broken lightpaths is not instantaneous, due to the delay of the tear down message propagation. In addition, nodes will not perceive the new resources instantaneously, because the slow convergence of IGP. Also, a BBM attempt may be slow, because the lightpaths will compete for the same resources, and in this case, the RSVP-TE may block setup attempts.

Our solution is the following: always when possible, the recovery of lightpaths will consider performing MBB. From viewpoint of the ingress node, MBB is possible if the recovery path does not contain any edge assigned by the Notify messages as "MBB unlikely" or "MBB impossible". The reuse of wavelengths in MBB is not mandatory, if the node finds a more advantageous recovery path that is disjoint with respect to the original path. For the lightpaths that do not satisfy this condition, BBM will be performed, and resources are released immediately after receiving the Notify message. When the ingress node is unable to find a candidate path to perform a recovery attempt, it will wait a random timeout (back-off), with a minimum safeguard time to receive updates from IGP.

The feasibility of MBB may change in a scenario where nodes are capable to perform wavelength conversion. In the scenario of Figure 1, the situation of lightpaths A and C will not change. For lightpath A it would still be possible to perform MBB recovery, and lightpath C would still have no wavelengths available in edges 3-4 and 5-6. However, the situation for lightpath B would change because now it is possible to perform a cross-connect of the MBB wavelengths to the edges 4-5 and 8-5. In this case, lightpaths A and B would perform MBB recovery and lightpath C would perform BBM recovery.

IV. PROPOSED SOLUTION

To provide some level of load balancing, the ingress node computes an explicit route to the destination using a load balancing heuristic. The most common heuristic consists in assigning a cost to a link that is proportional to the fraction of wavelengths in use with respect to the total number of wavelengths. The Weighted-Shortest-Cost-Path (WSCP) proposed by Hsu et al. [5] follows this strategy. We have modified WSCP to take into account the possibility of performing MBB, as defined in equation (1). A reduction factor (*mbbfactor*) is used to favor routes that reuse the wavelengths of the broken lightpath. In the formula, *hopweight* = *mbbfactor* if the edge contains a reusable wavelength and MBB is possible. Otherwise, *hopweight* = 1. In the expression, P_{sd} is the set of edges *e* connecting the source node to the destination node, and fw(*e*) and w(*e*) are, respectively, the number of free wavelengths and the total number of wavelengths in *e*. In the expression, *bfactor* controls the relative importance between assigning paths that contribute to load balance or are shorter in number of hops. Algorithm 1 is used to determine the recovery route and is responsible to make the decision to use MBB or BBM.

$$\Theta(P_{sd}) = \sum_{e \in P_{sd}} hopweight + \frac{\mathrm{fw}(e)}{\mathrm{w}(e)} \cdot bfactor \cdot \qquad (1)$$

In the following, we show how to build the recovery path using the route computed with the algorithm in Figure 6. According to RSVP-TE, the Path message may include a Label Set (*lset*) that restricts the range of wavelengths that can be selected by the downstream node. Nodes capable of performing wavelength conversion may expand the *lset*. The Path message may also include a Suggested Label (*sl*), a wavelength chosen from the *lset* that is preferentially offered to the downstream node. If the downstream node is able to use *sl*, it performs a cross-connect between the *sl* received from the upstream node and the *sl* offered to the downstream node. Once the Path message is received by the egress node, it selects the Generalized Label (*gl*) and transmits it upstream using the Resv message. If the *gl* is different from the *sl*, a node must remake the cross-connect with the *gl*. The Explicit Route Object (*ero*) permits to define the route and the wavelengths used along the path. The crank-back extensions introduce the possibility to fix blocked setup request without signaling a new setup request from the ingress node. Segment-based re-routing allows any upstream node that receives an error message to make a correction in the setup request through a new Path message.

Begin: The algorithm is triggered by the first Notify message received by the ingress node.

1. Save the information received in the first Notify message in: "failed", "MBB likely", "MBB unlikely" and "MBB impossible" edge sets.
2. Wait for the second Notify message. If message arrive before timeout, go to Step 3, otherwise go to Step 4.
3. Update the information received in the edge sets.
4. Create a graph including the edges that are not in the "failed" edges set; the edges not depleted according to IGP; and the edges depleted but with wavelengths used by the original broken lightpath. Depleted edges usable only with MBB waves are called "MBB only".
5. Compute a list of candidate paths by considering the k-shortest paths with respect to the number of hops.
6. Eliminate from the list of candidate paths all paths that include at least one "MBB only" edge and at least one "MBB unlikely" or "MBB impossible" edge.
7. If the remaining set of candidate paths is not empty, go to Step 9.
8. Tear down the original lightpath to free its resources, and perform a recovery attempt without explicit routes after a back-off timeout. Terminate the algorithm.
9. Select the best route among the candidate paths according to the cost function in equation (1). If the best route does not contain any edge with a MBB wavelength, tear down the original lightpath to free its resources. Perform a recovery attempt using the best route as an explicit route. Terminate the algorithm.

Figure 6. Algorithm 1: Determine the recovery route.

Figure 7 illustrates the sequence of messages required to setup a recovery lightpath using our method. The Path message is generate using an *ero* with explicit labels, associated to each hop in the *ero* object (see the *lero* sub-

 ISBN: 978-1-61208-318-6

object in the figure). The ingress node fills the *lero* with the MBB wavelengths that are present in the *ero*. New links in the recovery path have no explicit label (they are indicated as "0" in the *lero* sub-object). The ingress node also indicates the lightpath is being recovered using the Association Object (*ao*), informing that for the broken lightpath, a cross-connect can be undone and the reserved wavelengths can be released.

The Path planned by the ingress node may be unfeasible. In the example in Figure 7, the ingress node knows there is a free wavelength at the link 4-7. However, it does not know which wavelength it is, neither if the node 4 may perform a conversion to this wavelength. Because of this, node 4 generates an error, sending a PathErr message back to node 3. Instead of forwarding the error upstream, node 3 computes a new path to the destination excluding node 4. The new path is included in a new Path message as an explicit route and sent to the node 8 (re-routing). The Path message may also carry an excluded route object (*xro*), in order to inform to the downstream nodes known blocking resources. This is necessary if another node is required to solve a blocking by performing another segment re-route.

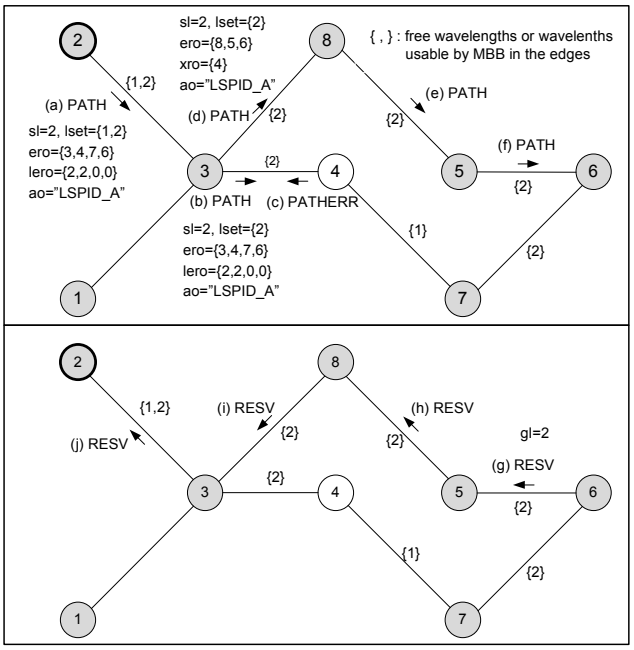


Figure 7. A Notify message is used to report the ingress node of each interrupted lightpath.

The algorithm in Figure 8 describes the procedure to determine the next node to forward a Path message. This procedure can be triggered by a Path message received from the upstream node, or a by PathErr message received by the downstream node. In the first case, some information such as the upstream label set must be retrieved from the node state database. The set of adjacent edges (local ports) with no wavelengths that satisfy the label set (*blockedPorts*) is computed using the local information of the node. The *xro* and the *nex* objects are specific crank-back information created by the node or received from the upstream node.

V. EVALUATION

We have developed a simulator for the GMPLS control plane using Wolfram Mathematica. The RSVP-TE messages propagate as individual packets and are delayed by the transmission rate, link propagation and queuing in the input and output ports of the nodes. The Reconfigurable Optical Add-Drop Multiplexer (ROADM) is able to process only one RSVP-TE message at a time. We have included in the simulator all elements required to estimate the setup time. The control plane messages propagate as individual packets and are delayed by the transmission rate (1Gbps), link propagation and queuing in the input and output ports of the nodes. The ROADM nodes are modeled as single processor entities, i.e., each node is able to process only one RSVP-TE/SDN protocol message at a time. Incoming messages are queued and processed sequentially in a FIFO. An optical cross-connect (i.e., the creation a flow in a WOFS) is the most timing consuming operation. The time to perform an optical cross-connect is 10 ms and to release a cross-connect, 5 ms. The time consumed to process Path and Resv messages is 2 ms. The time consumed to process PathErr, ResvErr, PathTear is 1 ms. Lightpaths are teared down explicitly.

Begin: the algorithm is triggered by a Path or a PathErr message.

1. If the node has no wavelength converter, determine *blockedPorts* as the set of adjacent edges with no free wavelengths included in *lset*. Otherwise, set *blockedPorts*= Ø.
2. If the procedure has been triggered by a Path message, and it includes an *ero*, determines next hop from it. If the edge connecting to the next hop does not belong to *blockedPorts*, go to step 8.
3. Determine the set of edges that has no more wavelengths available: *depletedEdges*.
4. If the procedure has been triggered by a PathErr message and *nex* is present, set *xro* with the nodes in *nex*. If *nex* is not present, set *xro* with the node that generated the PathErr message.
5. Build a graph excluding the edges in *depletedEdges* and *blockedPorts* and the *xro* nodes.
6. If the graph is connected, go to Step 7. Otherwise, go to Step 9.
7. Compute the recovery path from the current node to the egress node using the metric given by Equation (1), with *hopweight* = 1. Set *ero* (with no explicit labels) with the new recovery path.
8. Update the *ls* with the wavelengths that can be cross-connected from the incoming *ls* to the local port connecting to the next hop. If the next-hop has an explicit label in the *ero*, set the corresponding wavelength as the sl. Otherwise, selects a random wavelength from the ls as the next sl. Send the Path message to the downstream node, and terminate.
9. Send a PathErr message to the upstream node including itself in *nex*, and terminate.

Figure 8. Algorithm 2: Build and forward a Path message.

The topologies of the control plane and the data plane are identical. All links have 32 wavelengths and all nodes have a shared converter pool with capacity to perform 8 wavelength conversions. The parameters *mbbfactor* and *bfactor* in Equation (1) are set to 0.5 and 4, respectively.

 ISBN: 978-1-61208-318-6

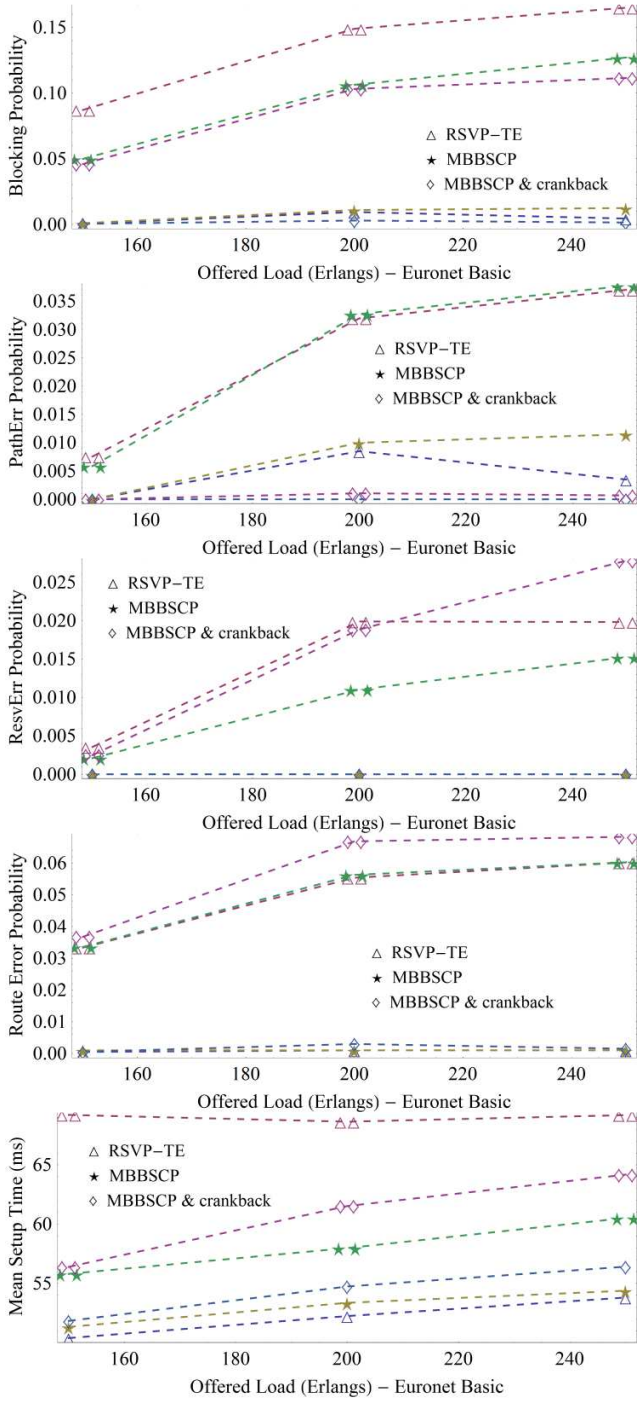


Figure 9. Evaluation of the Euronet basic topology.

The traffic load is generated as in most papers discussed in our review section. The network is submitted to a load of setup requests uniformly distributed among all pairs of nodes. The requests are controlled by two exponential variables: average interval among requests $1/\lambda$ and average duration of the lightpaths $1/\mu$. The total setup request load is measured in Erlangs λ/μ. To generate a variable setup request load we set $1/\mu = 2400$h, and we vary the value of λ. For each load scenario we performed the simulation for 2000 setup requests. The number of failures is variable, because they are based on the exponential failure rates computed taking into account the length of the links, and the number of optical amplifiers in the spans. Failures of nodes are perceived as multiple link failures by the adjacent nodes. Depending on the failure, or the importance of the link or node that failed, dozens of simultaneous recovery attempts may be performed simultaneously. In general, the number of the recoveries in the evaluated scenarios varied between 2000 and 3000. The amount of simulated connections resulted in a small standard deviation, of the order of 10^{-4} for the average probabilities and of the order of 10^{-1} for the mean setup time.

The following methods are evaluated: (i) RSVP-TE: standard distributed RWA. (ii) MBBSCP: uses explicit routes to support MBB, according to Algorithm 1. (iii) MBBSCP & crank-back: uses explicit routes to support MBB, and the crank-back re-routing, according to Algorithm 2. We present the results obtained for two distinct topologies, based on variations of the Pan-European network (basic and large) and the NFS network. For all networks, we have assumed that the topologies of the control plane and the data plane are identical. The control plane uses a reserved wavelength in all links with a throughput of 1 Gbps.

Figure 9 shows the obtained results for the Pan-European network basic topology. The performance metric is the average blocking probability. The blocking probability of the first setup is indicated as single plot markers, and the recovery blocking probability is indicated as double plot markers. In all scenarios, the best performance is obtained by the MBBSCP & crank-back approach, followed by the MBBSCP approach, indicating that the major influence results from the coordination of the recovery attempts. Blockages caused by the exhaustion of drop ports are not considered because it cannot be controlled by any of the methods. The reason for exhaustion is the variable load that can saturate drop ports on the ends of the connections. However, the failure to consider this effect does not affect the results, because it occurs in all methods evaluated.

There are basically three main reasons for a setup attempt not to be completed: (i) A PathErr, caused when a node is not able to find a wavelength in the downstream port that satisfies the incoming label set restrictions. (ii) A ResvErr, caused when the label selected by the downstream node cannot be used, because this label has been assigned to another lightpath since the Path message was forwarded. (iii) A route error, caused when a node cannot find a route to the egress node (caused by depleted edges or failed edges or failed nodes). The MBBSCP is expected to reduce the number of route errors. Crank-back is expected to reduce the number blocking caused by PathErr.

The number of recovery attempts varied from 2248 (lowest load scenario) to 2914 (highest load scenario). The number of recovery setups completed with the help of crank-back re-route increases consistently with the load of the network. It is insignificant for the lowest load (150 Erlangs), but achieves 3.5% at 200 Erlangs and 4.2% at 250 Erlangs.

 ISBN: 978-1-61208-318-6

In the highest load situation, crank-back re-route is required even to help completing the setup of 1.65% of the lightpaths when they are first provisioned. At the highest load, the proposed method has reduced to almost zero the number of blocked recoveries caused by PathErr messages. The setups are blocked mainly due to route error (6.8%) and Resv error (2.8%). The crank-back slightly increases the recovery setup time, because requires a higher number of messages to complete the setup.

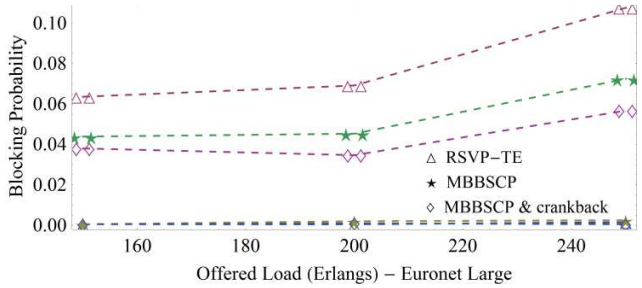


Figure 10. Evaluation of the Euronet large topology.

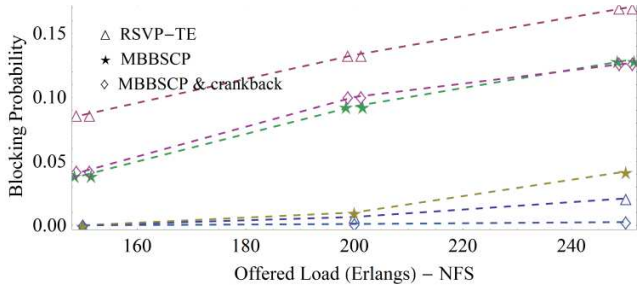


Figure 11. Evaluation of the NFS topology.

The results for the other topologies are similar to the Euronet basic, so we are going to present them briefly. For the Euronet large (see Figure 10) and for the NFS basic topology (see Figure 11), it can be observed that the recovery blocking probability steadily increases with the offered. Because these networks provide many recovery alternatives, the advantage of the proposed methods is more visible. Observe that the distance between the RSVP-TE approach and our proposed extensions increases with higher loads.

VI. CONCLUSION AND FUTURE WORK

In optical networks, the unplanned recovery technique based on re-routing poses a number of difficulties that are not observed in packet switched networks. In this paper, we have proposed a method to improve the robustness of lightpath setup performed in a distributed scenario. The proposed approach takes use of the flexibility provided by the Notify messages and crank-back extensions introduced by GMPLS. Our evaluations showed that our approach can significantly reduce the blocking probability of recovery attempts. However, there is still room for improvement. In special, with regard to the method proposed in this paper, the ResvErr messages have not been handled by crank-back. ResvErr occurrence is insignificant when the network is in a normal state of operation, but it is an important factor to be addressed during restoration, because the concurrence for resources may prevent a node to honor the wavelengths offered by the Path messages. We intend to address this issue by improving the crank-back re-route logic to also take into account this effect. We also intend to develop a method for dimensioning the network to give a degree of assurance about the success of lightpaths restorations.

REFERENCES

[1] N. Andriolli, J. Buron, S. Ruepp, F. Cugini, L. Valcarenghi, and P. Castoldi, “Label preference schemes in gmpls controlled networks,” Communications letters, IEEE, vol. 10, December 2006, pp. 849–851.

[2] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash, “Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE,” RFC 4920 (Proposed Standard), July 2007.

[3] A. Giorgetti, N. Sambo, I. Cerutti, N. Andriolli, and P. Castoldi, “Label preference schemes for lightpath provisioning and restoration in distributed gmpls networks,” Journal of lightwave technology, vol. 27, March 2009, pp. 688–697.

[4] A. Giorgetti, N. Sambo, I. Cerutti, N. Andriolli, and P. Castoldi, “Suggested vector scheme with crankback mechanism in gmpls-controlled optical networks,” In Optical Network Design and Modeling (ONDM), 14th Conference on, February 2010, pp. 1-6.

[5] C. Hsu, T. Liu, and N. Huang, “An adaptive routing strategy for wavelength-routed networks with wavelength conversion capability,” In Communications, ICC 2002, IEEE International Conference on, vol. 5, April 2002, pp. 2860–2864.

[6] J. Lang, B. Rajagopalan, and D. Papadimitriou. Generalized multiprotocol label switching (gmpls) recovery functional specification. RFC 4426 (Proposed Standard), March 2006.

[7] A. Manolova, et al., “Distributed sharing of functionalities and resources in survivable gmpls-controlled wsons,” IEEE/OSA Journal of Optical Communications and Networking , vol. 4, March 2012, pp. 219 –228.

[8] A. Manolova, et al., “Wavelengths and regenerators sharing in gmpls-controlled wsons,” In Global Telecommunications Conference (GLOBECOM 2010), IEEE, December 2010, pp. 1-5.

[9] A. Manolova, et al., “Shared path protection in gmpls networks with limited wavelength conversion capability,” In High Performance Switching and Routing (HPSR), 2010 International Conference on, June 2010, pp. 203–208.

[10] R. Munoz, R. Casellas, and R. Martinez, “An experimental signalling enhancement to efficiently encompass wcc and backup sharing in gmplsenabled wavelength-routed networks,” In Communications, ICC’08 IEEE International Conference on, May 2008, pp. 5401–5406.

[11] G.S. Pavani and H. Waldman. “Routing and wavelength assignment with crankback re-routing extensions by means of ant colony optimization,” IEEE Journal on selected areas in communications, vol. 28, May 2010, pp. 532 –541.

[12] N. Sambo, N. Andriolli, A. Giorgetti, and P. Castoldi, “Wavelength preference in gmpls-controlled wavelength switched optical networks,” Network protocols and algorithms, vol. 3, July 2011, pp. 110–125.

[13] Yue Chen, Nan Hua, Xiaoping Zheng, and Chunming Qiao, "Experimenting with immediate re-routing on an information-diffusion-based routing test-bed," Optical Fiber Communication Conference and Exposition (OFC/NFOEC), and the National Fiber Optic Engineers Conference , March 2011, pp.1–3.

[14] E. Jamhour, and M. Penna, "Evaluation of segment-based crankback re-routing for GMPLS-based WSON," Telecommunications (ICT), 20th International Conference on , May 2013, pp.1–5.

 ISBN: 978-1-61208-318-6

An Overview of Switching Solutions for Wired Industrial Ethernet

György Kálmán, Dalimir Orfanus, Rahil Hussain
ABB Corporate Research Norway
{gyorgy.kalman, dalimir.orfanus, rahil.hussain}@no.abb.com

***Abstract*—Industrial Ethernet is the preferred network technology in industrial green field deployments. Active devices interconnecting the nodes are switches. In an industrial deployment, nodes are typically located on the same Layer 2 network. Routers or firewalls are almost exclusively used at the network edges. The current trend on engineering of industrial devices is to include an embedded Ethernet switch, instead of using discrete units. This paper is giving an overview on switch implementation possibilities, with respect to performance, features, logical architecture and flexibility.**

Index Terms*—*industrial Ethernet, switch, embedded, discrete, soft switch, forwarding, performance, QoS

I. Introduction

Ethernet is already the dominating technology on the control and higher levels of an automation network and is expected to spread also into the field networks.

Because of resource constraints and Quality of Service (QoS) requirements, most of the automation networks are implemented as Local Area Networks (LANs). Although separating firewalls or routers are used between the automation network and the company network or the internet, inside the system, the network is typically interconnected on layer 2, by switches, as shown on figure 1.

The paper is structured as follows: the second section provides an background overview on industrial Ethernet, focusing on topologies and QoS. Then the possible architectural solutions are explained, with discrete, embedded and soft switches as main categories. Performance comparison is given based on our testbed measurements focusing on latency and jitter. A conclusion on possible fields of use for the discrete, embedded and soft solutions is given.

II. Industrial Ethernet background

Industrial Ethernet enables the use of standard Ethernet devices and the IP protocol suite in automation networks. By implementing networks based on Ethernet, vendors can create infrastructures, which provide improved bandwidth, resiliency and network security compared to fieldbus solutions (figure 1). As an additional value, the use of already established standards lowers the risk associated with technology development.

A number of issues arise from the fact that the Ethernet networks are replacing the fieldbuses. A heritage of the fieldbus past is the dominant use of bus-like topologies (figure 2 resulting in suboptimal operation of Ethernet [1], [6].

The most challenging topology type are long chains of switches (figure 2, which are often closed to rings. While

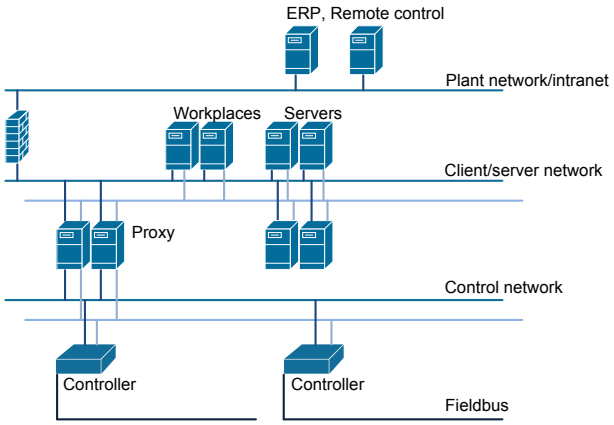


Fig. 1. An example of automation network architecture

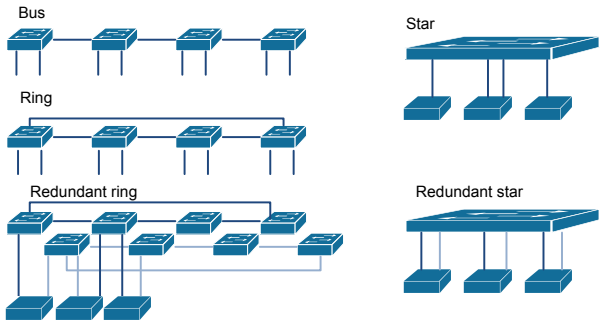


Fig. 2. Industrial Ethernet topologies

Rapid Spanning Tree Protocol (RSTP) was designed with loop-avoidance in mind, it is a widely used redundancy protocol in industry. In a ring structure, RSTP will disable one link and render the network into a special tree, a line of switches.

Although a line is a valid Ethernet topology and the technology will work, the industrial network will suffer from scalability issues at a much smaller end-node count than it could be expected from office experience [2]. The very long and sparse spanning tree (in practice only a single path) is having a low branching factor and can lead to excess latency and jitter [3], [7], [8].

In office environments, high port count switches are used to implement a high branching factor network, thus the issues associated with cascaded switches are less important [4], [5].

 ISBN: 978-1-61208-318-6

Also, in a typical setup, an office LAN is much earlier divided into subnetworks using firewalls and/or routers than reaching a deep spanning tree.

As an indirect result of the low branching factor and the pressure for lower costs and relatively high price of managed industrial Ethernet switches, the recent trend is to include low port count switches into the devices, so that devices can be interconnected into a cascade or a low branching factor tree without the use of external switches. This trend also shows that latency and jitter from the long sparse trees will be persistent in coming years and in combination with time-critical automation tasks, might be a limitation for the scalability of networks [9], [11].

Integrated switches are expected to lower the cost of building the network but without a penalty in features or QoS. In the following sections, we provide an overview of the typical embedded switch architectures and how they could be fitted into the industrial landscape.

III. Architecture possibilities

With a few exceptions, only managed switches are being deployed in industrial Ethernet networks. This is a result of the different requirements arising in the industrial environment compared to office networks.

A. Discrete switches

Unmanaged switches offer a low-price connectivity solution, where leafs are placed into the same network and the ingress traffic can be treated with the same rules independently of the port. In a direct comparison, unmanaged switches fail to meet the redundancy and logical segmentation requirements of the upper network layers of the automation network architecture.

Managed switches offer redundancy and loop-avoidance functions with, e.g., RSTP, logical segmentation using Virtual LANs (VLANs), remote management with Simple Network Management Protocol (SNMP) and troubleshooting features like port mirroring.

There are devices in the office networks, located between these two levels, called smart switches. They offer the most of the managed switch functions, but lack, e.g., SNMP management. Introducing a similar class of devices into automation networks might be of interest, since having a more grained approach on switch features can lead to a more cost-effective network architecture.

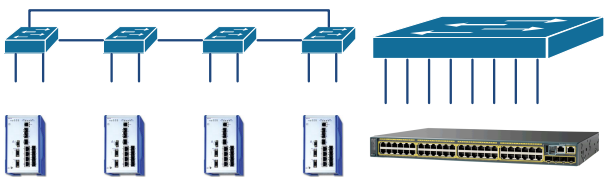


Fig. 3. Typical switch size comparison

There are few arguments against the use of discrete switches and most of them originate from the specific industrial landscape: the low branching factor, which results in a high number of low port-count switches, as shown on figure 3. The high number of standalone switches and the rugged hardware leads to a high per port price.

The low port count is even more apparent in the daisy-chained field networks, where Ethernet is also expected to replace the legacy communication solutions but typically it has to utilize the same topology.

In such environments, using the typical 8-10 port managed switches is rather expensive, as even these number of ports will not be utilized in addition to the higher management effort. To overcome price pressure, excessive engineering complexity and dependency on third party devices, vendors move towards embedded solutions.

IV. Embedded switches

Integrating a switch module into devices like controllers is on the agenda of automation vendors. These modules could take the tasks of discrete switches in the lower levels of automation networks. The construction of these units is potentially cheaper than using a separate switch, as, e.g., a low-end switch fabric, can provide a few gigabit/second of non-blocking bandwidth.

There are several important issues around the integration of devices. The first is the question of interface towards the host device.

The typical architecture offers an internal interface towards the host, which is implemented as a standard, but internal, Ethernet link. This setup is analog with the discrete switch case, only the interface connecting the host and the switch has been exchanged with the internal connection.

Switch modules by default only forward the traffic and all features, which are needed to implement a managed switch, have to be run in the host.

Another solution might be with an additional cost, to include a CPU on the switch module and implement the managed functions on the board, thus in practice be an independent managed switch inside the housing of the host.

These integrated modules are expected to deliver similar performance results as their low port count discrete counterparts and also to offer the similar range of services. If the management functions are implemented by using the CPU on the host, multicore platforms can be exploited by moving the forwarding-connected functions to one core and running the other functions on an other one, even on a different operating system if needed.

The possible drawback with these modules is that the host is still only connected with one internal port, which means that the host has no possibility to monitor the whole network traffic, if the aggregated bandwidth use exceeds the host's bandwidth.

A. Minimum acceptable service level

A non-conventional approach is to minimize the implemented features of the devices. Typical requirements state that the switches used should be managed, but the actual feature set is not defined. Currently, managed switches typically implement the whole feature set expected from a managed switch

 ISBN: 978-1-61208-318-6

(complying to IEEE 802.3D), but in most cases, only a handful of features are actually used and also out of these, some are enabled by using the default configuration (e.g., weighting of frames in QoS queues).

A reduction in both cost and management effort could be realized by implementing a set of minimum acceptable service-level switches. An office network approach is the smart switch device class, for example the NetGear JFS524e, where the feature set is restricted to offer easier management and cost reduction in hardware. The smart switches represent more a restricted managed switch, but in the industrial environment, an approach from the opposite direction, extending the features of an unmanaged switch might be more interesting.

The motivation to use such devices is partly supported by the reduced development and device cost, but more importantly, the special circumstances of industrial deployments are also supporting this solution. One of the more complex services of the discrete switches are connected to traffic manipulation and security functions.

The gain associated with, e.g., Internet Group Management Protocol (IGMP) is that it can reduce the link load by groping the receivers of multicast streams and also to protect other devices from using resources on traffic, which they have no use for. Protocols Multiple MAC Registration Protocols (MMRP) and Multiple Group Registration Protocol (MGRP), are also expected to reduce traffic load in areas where, e.g., a VLAN has no clients configured.

The other group of protocols are the network operation functions, e.g., RSTP, and IEEE 802.1X using Remote Authentication Dial In User Service (RADIUS). These protocols are run to keep the network loop-free and ensure network integriti and to allow secure authentication of new nodes.

Although all of these protocols are useful in an average network topology, in the industry-typical long and sparse trees, their gain is reduced. For increased traffic effectiveness: because of operational safety, networks anyway have to be designed so, that they can carry the whole network traffic, so the gain offered by grouping protocols might be limited. The main problem associated with grouping protocols in the typical line topology is, that the resources need to be reserved over the whole path if nodes are expected to join or leave on the ports. The traffic reduction efficiency for line topologies depend on the actual traffic type. For example, Multiple VLAN Registration Protocol (MVRP) might cut out some VLANs to be carried on a specific path, but if new nodes are allowed to join to a segment, the bandwidth for carrying additional or all of the existing VLANs shall be possible, thus the bandwidth spared by MVRP shall be reserved. In case of multicast protocols, like MMRP can be beneficial, but in this case also, at least the bandwidth need for all multicast groups shall be reserved even if not all of the groups are transmitted.

The execution of RSTP might also be of limited use, if the switches are organized in a chain and in every case, if the main uplink is broken, the other designated port towards the other switches will be chosen. Also, the topology of these networks is very static.

A possible solution is to use a compromise: deploy as simple as possible switches where chained topologies are used and include fully-featured discrete units where a tree connection structure is used (e.g., interconnecting rings or network backbone). Thus, the discrete units can run all the grouping protocols and reduce the load introduced to the ring, but inside the ring no further optimization is done. The simple devices shall be transparent on all protocols they do not support.

V. Soft switches

Embedded communication solutions are now allowing the implementation of a soft switch processing traffic of several gigabit/s of traffic on low consumption System on a Chips (SoCs). These are typically combined from an embedded CPU, a set of independent network controllers and a chipset, which integrates these into one system.

The positive point with these setups is, that the host is the switch: it is possible to monitor the whole traffic flow directly on the interfaces. Also, the platform can provide a good basis for feature extensions toward implementing a router, firewall or network monitoring appliances.

A high performance, multicore SoC can also serve as a platform for automation tasks and with the use of a multicore CPU, the communication and automation tasks could be run separated.

The main drawback of soft switches is the absence of the dedicated switching fabric. The throughput of the platform is prone to the actual implementation of the chipset, and used driver and operating system as well. Also the limitations of the bus system and the network interfaces are summed, which can lead to insufficient performance in a low latency environments. The price tag of such a solution can be justified if the device is utilized also in other tasks not only bridging.

VI. Feature comparison

The reviewed architectures show that if the switching solution is chosen, the future possibilities regarding traffic management, performance and feature set are being reduced.

Discrete switches offer high performance and a long list of management features and supported protocols. Embedded switch modules are implementing switching, but protocol and management features have to be implemented by the host or by a separate CPU and they only offer statistic multiplexing towards the host if utilized bandwidth exceeds what the host interface can carry. Soft switches are in practice implementing the embedded switch scenario but without the hardware switch module, thus while offering full access to all traffic crossing the interfaces, they also suffer from the largest delays.

From the forwarding performance side, for large port counts, discrete switches offer the best solution, since a high-speed, non-blocking backplane is a hard requirement in this area. For smaller and medium sized switches (4-16 ports), an embedded solution can also be viable, as for low port count even the cheaper backplane solutions can provide enough bandwidth. It

 ISBN: 978-1-61208-318-6

is also less probable, that such a switch will be experiencing a situation where all of the ports are fully utilized.

Our measurements on the forwarding latency and throughput of switches showed marginal differences between discrete and embedded solutions while the tests executed on the soft switch platform resulted in weaker performance figures.

VII. Performance Measurement

A. Measurement and test equipment

Our test was implemented with the use of an array of discrete managed switches. The traffic generator was a Softing Industrial Ethernet Tester (OEM Psiber LanExpert 80), which can generate traffic between its two gigabit Ethernet interfaces and was acting as traffic source and sink.

Our tests were split into two areas: one was to measure the latency between two ports of the same switch to provide a way to compare the raw performance. The second area was to show switched Ethernet behaviour in a typical industrial setup, where switches are chained and the ingress and egress links are only 100Mbps while inter-switch links are 1Gbps. The initial results based on the LanExpert measurements showed no significant difference in latency or throughput between the embedded and discrete units.

To measure the latency between 100Mbps endpoints, we need preciseness ideally at the level of a bit-duration or better, which is 10ns for the Fast Ethernet. Since the LanExpert's measurement capabilities were not satisfactory for generation of the statistics and exact measurement of forwarding behaviour in a cascade, we decided to use the EtherCAT network consisting of the master (P2020 board) and two slaves (figure 4. EtherCAT provides service called *Distributed Clock* (DC), which can precisely synchronize clock in slaves with time resolution of 10ns and has dedicated hardware in slaves to measure network latency.

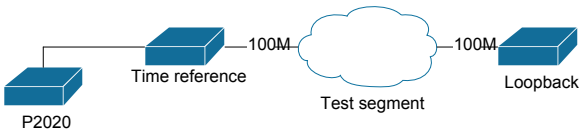


Fig. 4. Testbed setup

To assess the performance of the selected equipment and to be able to provide guidelines for network planning, we set up the following measurements:

B. Default forwarding latency

Measurement of the time it takes for the frame to traverse the switch. It is composed from store and forward latency (L_{SF}), the switch fabric latency (L_{SW}), the wireline latency (L_{WL}) and the queuing latency (L_Q) [10].

L_{SF} depends on the frame length. The results are expected to show a linear growth of the latency with the longer frames [12].

Our architecture related measurement scenarios deals with latency between endpoints (both with 100Mbps) of serial connected switches and without any additional interfering traffic

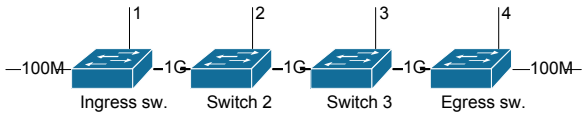


Fig. 5. Test segment setup

(see Figure 5). The purpose is to see the raw latency scaling of a network built by a chain of switches. We have four scenarios, each of them consisting with 1 up to 4 switches. Switches are between themselves connected with 1Gbps link. Initial measurements showed, in accordance with the LanExpert measurements, no significant difference between the discrete and embedded units, so the testbed was created by using 4 RuggedCom RS940G switches.

C. Standalone forwarding

Latencies and throughput between two interfaces of the same switch was measured with the LanExpert device and the results showed no significant difference between the capabilities of the embedded or the discrete units.

Measurements were performed on switches, which represent a significant part of the market: RuggedCom RS940G, Hirschmann RSR30, Moxa EDS-G509, a board based on Marvell 88E6352 switch chip and a soft switch using a stock Ubuntu linux and an Intel Xeon CPU with four chipset-integrated gigabit Ethernet interfaces. As a control, a test was also executed on a Cisco SG 200 switch (approximately the same performance class as the tested industrial variants), where differences in the results were also insignificant compared to the industrials. The measured latencies of the Marvell module are marginally lower, than the discrete counterparts, which is expected to be the result of the simpler architecture, as the module in the tested form implements only an unmanaged switch.

TABLE I
Throughput in k frames per second for respective frame sizes using 1 Gbps links

Frame size	RS940G	RSR30	EDS-G509	88E6352	soft
64	1481	1485	1485	1485	179
128	840	842	842	842	178
256	452	452	452	452	178
512	234	234	234	234	166
1024	119	119	119	119	119
1280	96	96	96	96	96
1518	81	81	81	81	81

The only considerable difference could be observed with the soft switch platform. It was not expected to hold the same latency figures but the maximal frame frequency of approximately 180kfps is low compared to the rest of the devices (table I. Although the latency is also higher (table II, the figures stay mostly within acceptable range for the majority of networking tasks. The low throughput observed with shorter frames in contrast, limits the specific setup's usability since it will not be able to utilize the bandwidth in case of a setup like our test segment, where two interfaces need to carry the

 ISBN: 978-1-61208-318-6

TABLE II
LATENCY IN MICROSECONDS FOR RESPECTIVE FRAME SIZES USING 1 GBPS LINKS

Frame size	RS940G	RSR30	EDS-G509	88E6352	soft
64	5	5	5	3	16
128	5	5	5	4	16
256	6	6	6	5	18
512	8	9	8	7	33
1024	12	13	12	11	114
1280	14	15	14	13	93
1518	16	17	16	15	99

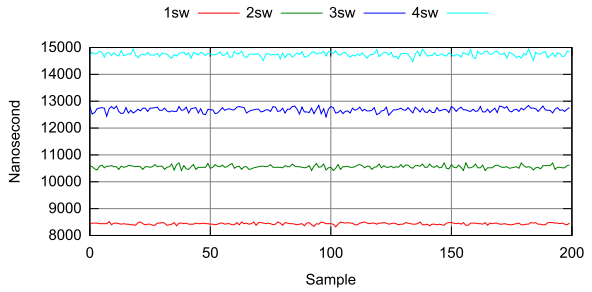


Fig. 6. Scenarios 1-4

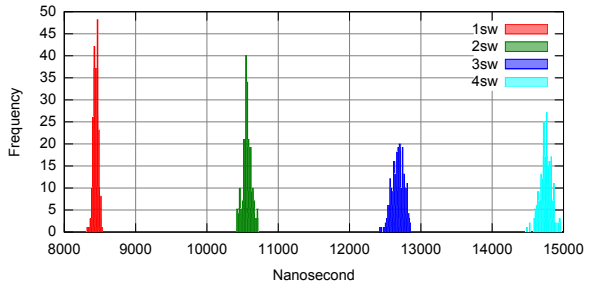


Fig. 7. Histogram Scenarios 1-4

aggregated traffic. It is expected that with different operating system and driver optimizations, better performance can be achieved.

D. Scaling of latency in a chain

Our measurements using the testbed extended with a variable cascade of switches (figure 5 show, that the discrete switches scaled as expected. When no additional traffic was injected to the measured interfaces, the latency growth was linear with minor variations. Measurements using an industry-typical scenario with 100 Mbps edge links and 1 Gbps internal links were executed. Four scenarios were measured, compromising of chains of 1-4 switches (figure 6).

Also the histogram on figure 7 shows the expected behavior: the longer the chain is built, the wider is the range of latencies measured. The determinism of the switching solutions can be seen on the measurements and that in low traffic installations a linear growth of latency can be expected.

The histogram of the measurements showed the expected result, with having the most step-like distribution at using one switch and a still narrow but more wide distribution of frame latencies for longer switch chains.

VIII. CONCLUSION

Our review shows that with regard to forwarding performance and latency, embedded switching solutions present a competitive solution compared to discrete units. Although, the offered set of features might differ and for managed functions, either the host CPU or an additional CPU for the switching board needs to be used, the performance expectation can be the same.

Soft switching on the other hand might be problematic when using a non-real time operating system and non-optimized drivers. If the software selection would move towards these, on the other hand, the flexibility of the platform would be limited. Our measurements showed that soft switches might be too slow to be used in a chained topology, but might be applicable in cases, where additional processing is required, for example as a controller with several network interfaces.

Our conclusion is, that if there are no clear requirements for traffic monitoring capabilities exceeding the bandwidth of the host-switch module link, embedded switches are a viable and effective solution for low branching factor industrial networks.

Soft switches are a viable solution for implementing routers or other network functions, where the additional latency compared to the other switching solutions is not critical as the processing of the data on higher layers will contribute to more latency and jitter as the switching.

REFERENCES

[1] F. Cen, T. Xing and K. Wu, Real-time Performance Evaluation of Line Topology Switched Ethernet, International Journal on Automation and Computing, October 2008, pages 376–380.
[2] N. Kakanakov, M. Shopov, G. Spasov and H. Hristev, Performance Evaluation of Switched Ethernet as Communication Media in Controller Networks, International Conference on Computer Systems and Technologies - CompSysTech 2007, Article No. 30.
[3] G. Marsal, B. Denis, J. Faure and G. Frey, Evaluation of Response Time in Ethernet-based Automation Systems, Conference on Emerging Technologies and Factory Automation 2008, pages 380–387.
[4] A. Manita, F. Simonot and Y. Song, Multi-Dimensional Markov Model for Performance Evaluation of an Ethernet Switch, Research report at INRIA, no. 4813, 2003.
[5] K. Beretis, I. Symeonidis, Experimental evaluation of end-to-end delay in switched Ethernet application in the automotive domain, International Conference on Computer Safety, Reliability and Security, Toulouse 2013.
[6] J. Jasperneite, P. Neumann, Switched Ethernet for Factory Communication, Conference on Emerging Technologies and Factory Automation 2001, pages 205–212.
[7] H. Lim, L. Völker and D. Herrscher, Challenges in a Future IP/Ethernet-based In-Car Network for Real-Time Applications, Design Automation Conference 2011, pages 7–12.
[8] T. Skeie, S. Johannessen and O. Holmeide, Timeliness of Real-Time IP Communication in Switched Industrial Ethernet Networks, IEEE Transactions on Industrial Informatics Vol.2, No.1, February 2006, pages 25-39.
[9] A. Jacobs, J. Wernicke, S. Oral, B. Gordon and A.D. George, Experimental Characterization of QoS in Commercial Ethernet Switches for Statistically Bounded Latency in Aircraft Networks, Report, High Performance Networking Group, University of Florida
[10] RuggedCom, Latency on a Switched Ethernet Network, 2008.
[11] Hirschmann, Real Time Services (QoS) In Ethernet Based Industrial Automation Networks, 2000.
[12] J. Georges, T. Divoux and E. Rondeau, Network Calculus: Application to switched real-time networking, ICST 2011, pages 399–407.

 ISBN: 978-1-61208-318-6

Experimental Analysis of TCP Behaviors against Bursty Packet Losses Caused by Transmission Interruption

Weikai Wang, Celimuge Wu, Satoshi Ohzahata, Toshihiko Kato
Graduate School of Information Systems
University of Electro-Communications
Chofu-shi, Tokyo, Japan
e-mail: *ohigai@net.is.uec.ac.jp, clmg@is.uec.ac.jp, ohzahata@is.uec.ac.jp, kato@ is.uec.ac.jp*

***Abstract*— Although TCP was originally designed to provide the reliable data transfer over the Internet, packet losses detected in TCP are considered as an indication of network congestion due to the high quality of data transmission provided by recent transmission technologies and media access control technologies. However, packet losses can be caused by transmission interruptions such as handoffs in the mobile networks and protection switching in the transport networks. These packet losses are bursty because the transmission interruptions continue for tens of miliseconds through several seconds. In this paper, we describe the experimental analysis of TCP behaviors by inserting errors such that all packets are lost during transmission interruptions. We have tested various TCP versions including those in Linux, that in Windows and that in Mac OS. This paper suggests (1) that the tested TCPs in Linux follow the similar procedure and retransmit lost packets quickly, (2) that TCP in Windows also behaves well but the increase of congestion window seems to be limited, and (3) that TCP in Mac OS has shown some problems in retransmitting contiguously lost packets.**

Keywords-TCP; Transmission Interruption; Bursty Packet Losses; Retransmission; SACK Based Loss Recovery.

I. INTRODUCTION

Transmission Control Protocol (TCP) is widely used as a transport protocol for the reliable data transfer. TCP recovers from packet losses by retransmitting lost packets and guarantees that the information sent is safely delivered to the receivers. But, recent transmission technologies and media access control technologies provide high quality of data transmission, and so, packet losses detected in TCP are considered as the indication of network congestion.

Although the possibility of packet losses caused by random bit errors is extremely low, it is possible that data are lost due to transmission interruptions. For example, packets will be lost during a handoff among base stations in the 3rd generation mobile telecommunication networks [1]. Similar packet losses occur during a channel switch in the protection switching systems [2], [3].

These packet losses are bursty, because such a transmission interruption continues in the order of tens of miliseconds through several seconds. TCP, of course, has the functionality to recover from those bursty packet losses, but it seems that the research activities on TCP performance focus on the congestion control scheme during light congestion situation where the number of lost packets is limited [4].

This paper describes the results of experimental analysis of TCP behaviors when a TCP data transmission suffers from bursty packet losses during a transmission interruption. We have tested several TCP versions; TCP implemented in the Linux operating system [4], TCP in the Windows 7 operating system, and TCP in the Mac OS X operating system. For those TCP versions, the TCP communication traces are examined in detail. As a result, we suggest that
(1) the tested TCPs in Linux follow the similar procedure and retransmit lost packets quickly, that
(2) TCP in Windows 7 also behaves well, but the increase of congestion window seems to be limited compared with those in Linux, and that
(3) TCP in Mac OS X sometimes takes longer time than the others to retransmit the packets lost during a transmission interruption.

So far, there have been some papers published focusing on the TCP behaviors against packet losses [5] – [7]. In [5], TCP over a 3G wireless system, IS2000, is discussed. Especially, it describes the periodical data transmission timing in IS2000 and its impact on TCP, and the effectiveness of selective acknowledgment (SACK) [8] and timestamp TCP options. In [6], TCP performance over commercial WiMAX-based network is presented. It compares New Reno, Cubic [9], Vegas and Veno TCP variants in terms of throughput, rount-trip time and retransmission rate, and points out that a WiMAX link is not well-suited for the aggressive Cubic and window auto-tuning. Zhu and Bai [7] compared the performance of Tahoe, Reno and SACK TCP when multiple packets are dropped, and shows that Reno suffers from performance problems at multiple drops while SACK works well. On the contrary, this paper gives the detailed packet level analysis of TCP behaviors against burst errors using the timeline charts and points out the problems in Mac OS X TCP which are not discussed in the other papers.

The rest of this paper consists of the following sections. Section 2 specifies the conditions of the transmission interruption test. Section 3 gives the results of various TCP versions. Section 4 describes a packet level behavior analysis for the results of TCP Reno in the Linux operating system and Mac OS X TCP. Section 5 gives the conclusions of this paper.

 ISBN: 978-1-61208-318-6

II. TEST CONDITIONS

Fig. 1 shows the configuration of the experiment. The TCP program to be tested is implemented in a personal computer (PC under test in the figure). It is connected to a wireless LAN (IEEE 802.11g) through an access point (AP), which is connected to the bridge emulating transmission interruptions through Gigabit Ethernet. The bridge injects a 200 milisecond interruption at every five second. During the interruption, the bridge discards all packets transferred in both directions. The bridge is connected the ftp server through Gigabit Ethernet.

The TCP communication is traced using tcpdump. The trace is taken in the PC under test and ftp server, and two traces are examined for each experiment.

In this test, PC under test works as an ftp client and sends a 10 megabyte file to the ftp server. The specification of the equipment is listed in Table I.

The TCP versions adopted in this test are as follows:

- TCP Reno: a traditional additional increase and multiplicative decrease (AIMD) control of congestion window with fast recovery.
- Cubic TCP [9]: congestion window control as a cubic function of time elapsed since a last congestion event. It has been the default of Linux TCP suite since 2006.
- TCP Westwood [10]: designed for wireless network by estimating the available bandwidth from ACK arrival intervals.
- TCP in Windows 7: default TCP in the Windows 7 operating system. It is said to combine slow and scalable way in the congestion window calculation (compound TCP [11]).
- TCP in Mac OS X: default TCP in the OS X operating system.

The configuration of TCP options, such as whether to use the window scale option and the SACK option or not, follows the default setting of the individual operating systems.

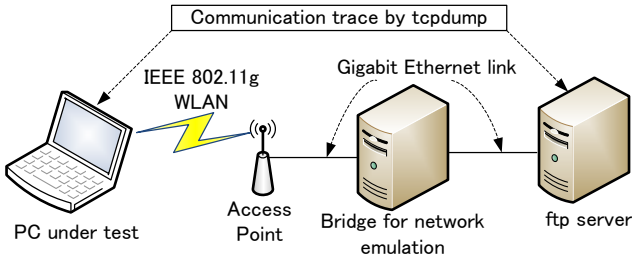


Figure 1. Configuration of Experiment

TABLE I. SPECIFICATION OF EQUIPMENT

Node	OS	Hardware specification
PC under test	Linux (Ubuntu 10.04)	Centrino2 CPU (2.53GHz) and 2GB memory
	Windows 7	
	Mac OS X (10.7.5)	MacbookPro with Core i7 CPU (2.4Ghz) and 8GB memory
bridge	Linux	Pentium4 HT CPU (2.4 GHz) and 2GB memory
ftp server	Linux	Core i5 CPU and 8GB memory

III. RESULTS OF EXPERIMENTS

We executed several test runs for each of TCP versions. This section shows typical results for those test runs as the graphs plotting the sequence number of transferred data segments (transferred bytes) versus the elapsed time since the SYN segment. The graph is generated by Wireshark [12] using the trace captured at the PC under test.

Fig. 2 shows the result of TCP Reno. The figure shows four discontinuous sections in the increase of sequence number. Three of them are labeled as "Reno (1)," "Reno (3)" and "Reno (4)." It is considered that they are caused by bursty packet losses injected at the bridge. For confirmation, Fig. 3 shows the similar graph generated from the trace captured at the ftp server side. Fig. 3 shows that the increasing status of the sequence number at the server side is similar with that at the PC under test side, and that there are parts where packets are lost contiguously in the four discontinuous sections.

There are two types of discontinuous sections in Fig. 2. One is the type for the first through the third sections. In this type, packets are lost in the middle of a continuous data sending in the TCP flow control. There is no time lag between the normal data transmission and the retransmission.

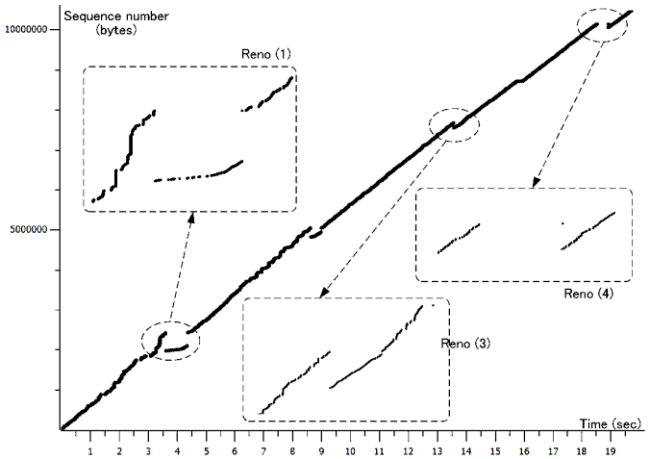


Figure 2. Sequence number vs. time for TCP Reno

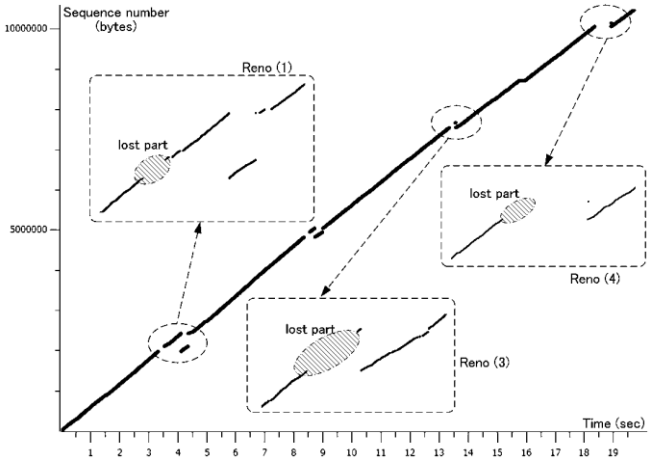


Figure 3. Sequence number vs. time for TCP Reno in ftp server side

The other type is that for the fourth discontinuous section (Reno (4)) in the figure. Packets are lost in the end of continuous data sending and there is a time lag before the retransmission starts. We will examine the packet level behaviors of these two types in the next section.

Fig. 4 shows the result of Cubic TCP. The sequence number versus time graph is similar with that of TCP Reno. There are four discontinuous sections; the first, and the second (Cubic (2)) and the fourth ones are of type with time lag, and the third one (Cubic (3)) is of type without time lag.

Fig. 5 shows the result of TCP Westwood. The graph is similar with those of TCP Reno and Cubic TCP. As the results of our experiment, it can be said that the TCP versions in the Linux operating system behave similarly for bursty packet losses, although they have different congestion control mechanisms.

Fig. 6 shows the result of TCP in Windows 7 operating system. This graph is also similar with those of TCP versions in the Linux operating system. But, there are only discontinuous sections with type of time lag. Besides the experiment described in Fig. 6, we executed three runs of the experiment and obtained the result that all the discontinuous sections are of type with time lag.

The reason for this difference is analyzed as follows. For the discontinuous section without time lag, the data sending needs to continue longer than the transmission interruption injected by the bridge, i.e., 200 miliseconds. So, the window size (advertised window and congestion window) needs to be large enough to allow PC under test to keep sending data segments. So, we have checked the advertised window size in the TCP Reno case and the Windows case and show the result in Fig. 7. In the case of TCP Reno, the advertised window goes to 451,840, but it goes to only 55,488 in the case of Windows TCP. It should be noted that the window scale option is used in both cases and that it is possible to specify a large window size. Generally, TCP receiver adjusts its window size dynamically to twice of the congestion window size which it estimated. This is called auto-tuning or dynamic right sizing [13]. So, it is considered that, in this experiment, the ftp server (receiver) estimated the congestion window of Windows 7 TCP much smaller than that of Linux TCP, and that Windows 7 TCP did not continue data sending longer than the 200 milisecond transmission interruption. So, all the discontinuous sections were of type with time gap. However, Windows 7 TCP also behaves well and recovers quickly from the bursty packet losses caused by transmission interruption.

In the end, Fig. 8 shows the result of TCP in Mac OS X.

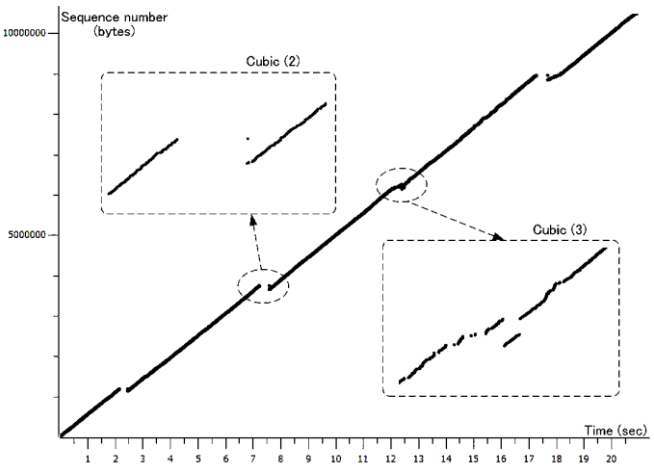


Figure 4. Sequence number vs. time for Cubic TCP

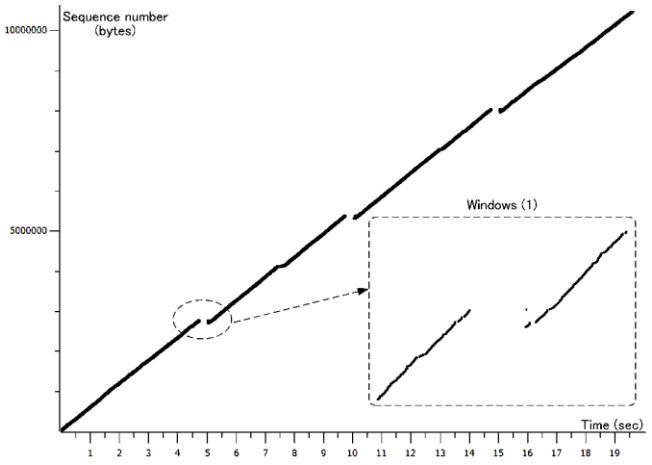


Figure 6. Sequence number vs. time for TCP in Windows

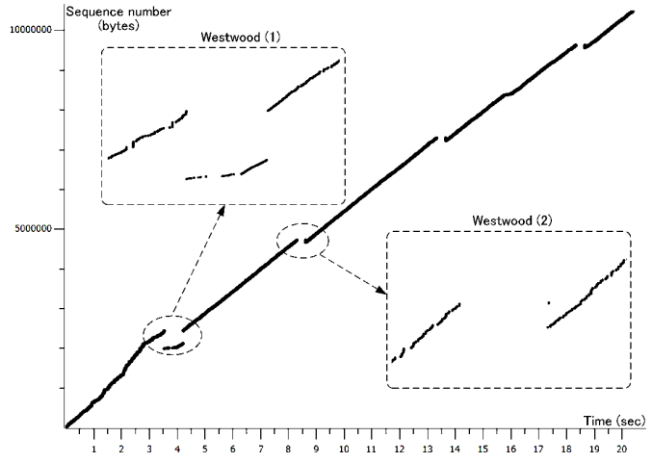


Figure 5. Sequence number vs. time for TCP Westwood

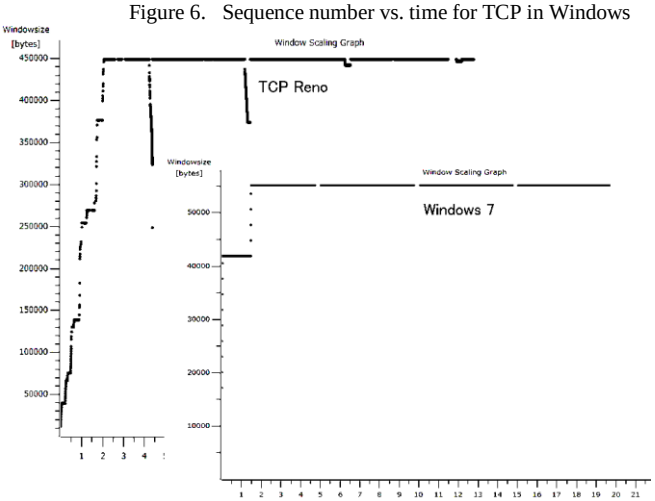


Figure 7. Advertized window for TCP Reno and Windows

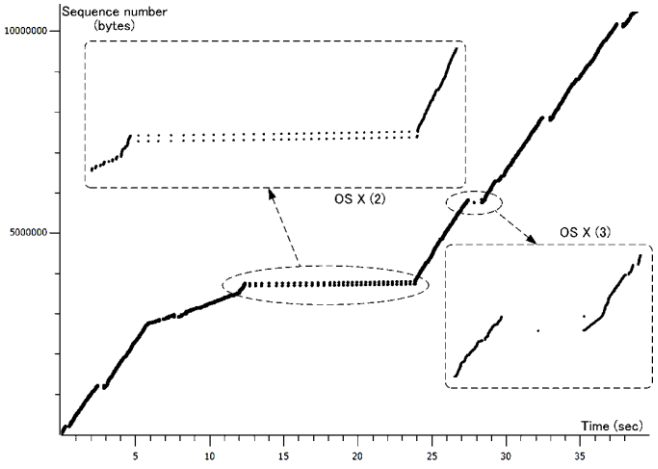


Figure 8. Sequence number vs. time for TCP in OS X

The graph is different from those of other TCP versions. In the second discontinuous section (OS X(2)), the graph is flat, i.e., the data sending rate is low, and data segments are sent intermittently. This section takes more than ten seconds and is considered to be worse error recovery than the other TCP versions described above. The detailed packet level analysis is given in the next section.

IV. DETAILED ANALYSIS OF TCP TRACES

A. *Packet Level Analysis of TCP Reno's First Discontinuous Section*

Fig. 9 shows the timeline of segment exchanges at the discontinuous section of Reno (1). This section is of type without time lag. The figure is described based on the trace at the PC under test side.

The data and ACK segments are specified according to the text format of tcpdump. For a data segment, the figure uses a representation such as "1967433:1968881(1448)," which means that the sequence number specified in the header of this segment is 1967433 (relative value from the SYN segment) and the number of bytes in this segment is 1448. The number 1968881 is the sequence number assigned the last byte in this segment plus 1, i.e., the sequence number in the header of the next data segment. For an ACK segment, this figure shows the acknowledgment number in the style of "ack 1965985" and, in addition, the other parameters such as a SACK option are also specified.

In this discontinuous section, 88 data segments are lost during the transmission interruption (*9(a)* in the figure). After that, the next data segment 2093409:2094857(1448) is sent to the ftp server (*9(b)*). In response to that, the ftp server returns an ACK segment 1967433 (SACK2094857-2096305) (*9(c)*). This ACK segment says that the sequence number of the next data which the receiver expects is 1,967,433, and that the receiver has received data from 2,094,857 to 2,096,304 [8]. Responding to this SACK segment, PC under test retransmits the data from sequence number 1,967,433. This is retransmitted because the ACK segment with SACK option says that the receiver received all of data up to 1,967,432 and that, in addition, it has

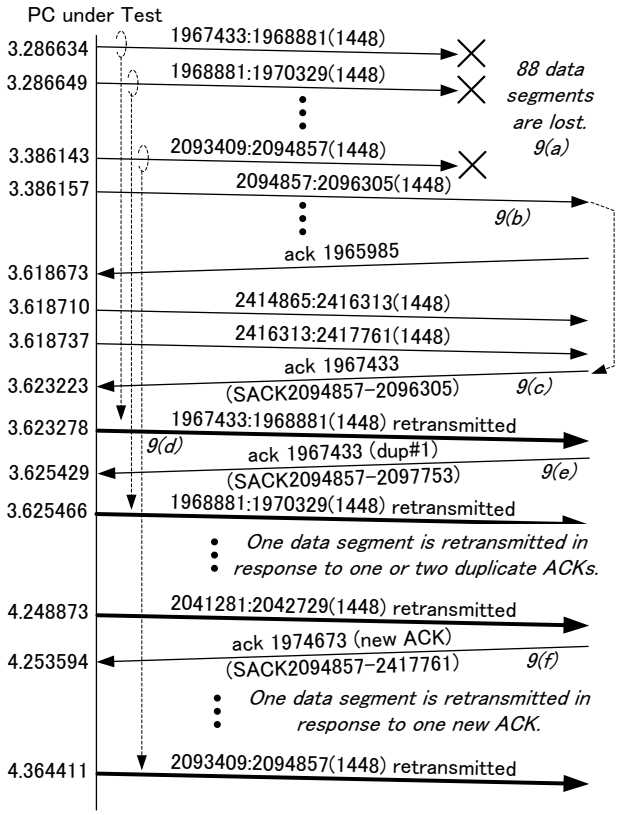


Figure 9. Timeline of segment exchanges at Reno (1)

received some SACKed data segments [14]. This means that it is possible that data from 1,967,433 to 2,094,856 are missing.

Before PC under test receives this ACK segment, it sends data segments up to 2416313:2417761(1448). Responding to those data segments, the ftp server returns other ACK segments with acknowledgment number set to 1967433 with SACK option whose range is increasing incrementally. They are duplicate ACKs with SACKs, and invoke the retransmission similarly with ACK *9(c)*.

The ftp server returns another ACK segment (*9(e)*) invoked by the next data segment, 1970329:1971777(1448). As described above, the information on newly received data is included in the SACK option (Please confirm that the range of SACK option becomes wider than that in ACK *9(c)*). After receiving all original data segments up to 2416313:2417761(1448), the ftp server receives retransmitted data segments, and responds to them by sending ACK segments (new ACKs) one by one (e.g., *9(f)*). Since these ACKs include SACK options, PC under test retransmits the next unacknowledged data segment one by one, until all the lost data segments are retransmitted.

In summary, at the discontinuous section without time lag, the receiver retransmits data segments just after it receives the first ACK with SACK option indicating a missing data gap. The retransmission seems to be based on the SACK. The requirement for this type of section is that the sender has a window size large enough to send data longer than the period of a transmission interruption. Similar

 ISBN: 978-1-61208-318-6

ones can be found in Reno (3), Cubic (3), Westwood (1), and so on.

B. Packet Level Analysis of TCP Reno's Fourth Discontinuous Section

Fig. 10 shows the time line of segment exchanges at the discontinuous section Reno (4) in Fig. 2. This section is of type without time lag.

In this discontinuous section, 58 data segments and one ACK segment are lost during the transmission interruption (*10(a)* in the figure). A data segment 10058857:10060305 (1448) is delivered to the ftp server, but the correspondent ACK segment (ack 10060305) is lost. The interruption losses up to the last data segment in continuous data sent in one TCP window. So, after an interruption, PC under test has any segments to send, and it just waits. When the retransmission timeout period passes, the sender retransmits the oldest unacknowledged data segment (*10(b)*).

Then, the receiver receives this segment, and it returns a corresponding ACK segment. But, the receiver already received this data segment before. In order to inform the sender of the duplicate receipt of this data segment, the receiver specify the duplicate range in the first block of the SACK option, as shown in the figure (*10(c)*). This mechanism is called DSACK [15].

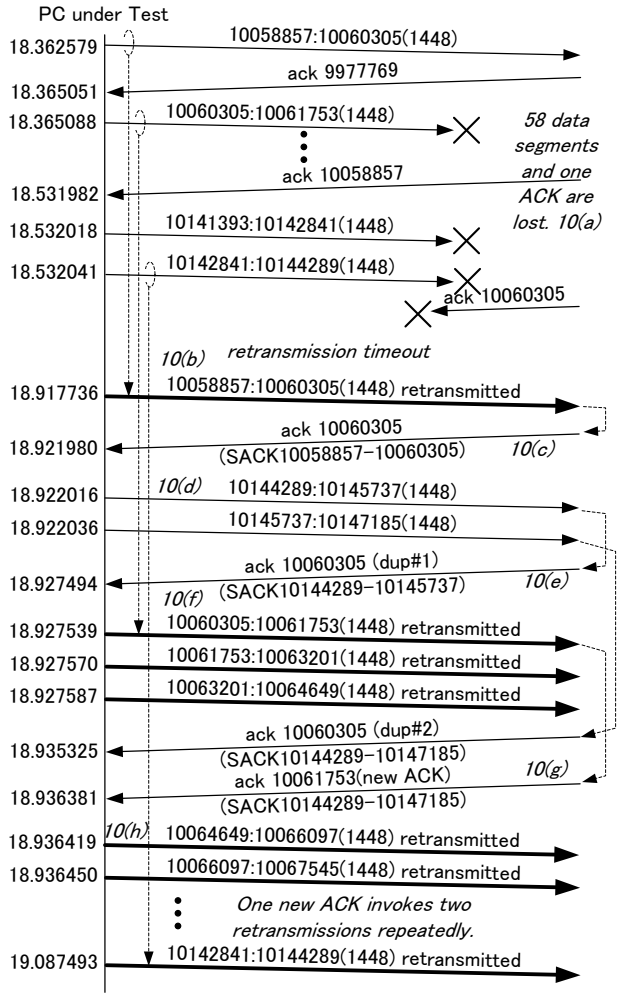


Figure 10. Timeline of segment exchanges at Reno (4)

After receiving this ACK segment, the sender retransmits two data segments following the last data segment which it has sent (*10(d)*). For each of these data segments, the receiver responds with an ACK segment, which will be a duplicate ACK with a SACK option (*10(e)*). When PC under test receives this ACK segment, it retransmits data segments (*10(f)*). This is considered as the retransmission by SACK option, which was also used in Fig. 9. These retransmissions generate new ACKs with the SACK option (*10(g)*), and again, they introduce the retransmissions (*10(h)*).

In summary, at the discontinuous section with time lag, the receiver will start to retransmit data segments due to the retransmission timeout. But, after the first retransmissions, the continuing retransmissions are invoked by the SACK based recovery. Similar discontinuous sections can be found in Cubic (2), Westwood (2), Windows (1), and so on.

C. Packet Level Analysis of Mac OS X TCP's Second Discontinuous Section

The discontinuous section OS X (2) is different from the others obtained in this experiment. Fig. 11 shows the time line of segment exchanges at this section.

At first, 46 data segments are lost during the transmission interruption (*11(a)* in the figure). Similarly with Fig. 10, the interruption losses up to the last data segment in one TCP window. So, the retransmission timeout occurs and the oldest unacknowledged data segment, 3675809:3677257 (1448), is retransmitted (*11(b)*). Then, the receiver returns a corresponding ACK segment for this data segment. It is a new ACK segment without any SACK options (*11(c)*). After receiving this ACK, the sender transmits a (new) data segment following the last data segment it sent (*11(d)*). For this data segment, the receiver responds an ACK segment which will be a duplicate ACK with a SACK option (*11(e)*).

So far, the timeline is very similar with that of Reno (4). But, when PC under test receives this ACK segment, it does not retransmit any data segments immediately. That is, the retransmit by the SACK option is not invoked. Instead, the sender waits for the retransmission timeout period and retransmits the oldest unacknowledged data segment (*11(f)*).

In the timeline, this sequence, a timeout retransmission, a new ACK, a new data, a duplicate ACK, and another timeout retransmission, is repeated. So, the intermittent data sending occurs. The reason for this sequence is considered to be the fact that the SACK based retransmission does not work well.

However, in the end of this sequence, PC under test receives a new ACK with SACK option (*11(g)*), and it retransmits next unacknowledged data segment (*11(h)*). In this part, it seems that the SACK based loss recovery works well. At another discontinuous section, OS X (3), the behavior of type with time lag is observed. Here, it seems that the SACK based retransmission is working.

In summary, Mac OS X TCP shows an intermittent type discontinuous section for bursty packet losses in a transmission interruption. The reason is that the loss recovery based on the SACK option does not work well. However, the SACK based loss recovery works in another

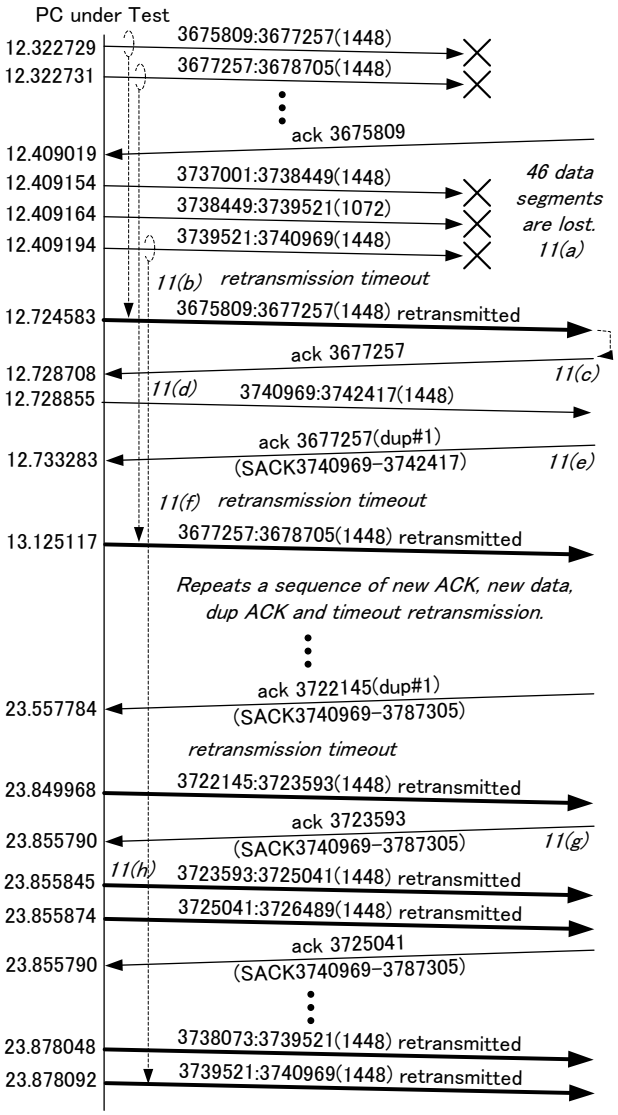


Figure 11. Timeline of segment exchanges at OS X (2)

discontinuous section in Mac OS X TCP, so we cannot say that it is not implemented in the Mac OS X.

V. CONCLUSIONS

This paper described the results of experimental analysis of TCP retransmission behaviors against bursty packet losses caused by transmission interruptions. We focused on several TCP versions; TCP Reno as a standard congestion control, Cubic TCP as a high speed version, TCP Westwood for a wireless network, TCP in Windows 7 and TCP in Mac OS X. The packet level detailed analysis for the TCP communication traces found the followings.

(1) The tested TCPs in Linux seem to follow the similar procedure and retransmit lost packets quickly. They behave as recovery types with time lag and without time lag, depending on whether timeout retransmission is used or not for the first missing data segment.

(2) TCP in Windows 7 also behaves well, but discontinuous sections caused by transmission interruptions are with type of time lag. The reason seems to be that the increase of congestion window of Windows 7 TCP much smaller than that of Linux TCP, and that the receiver does not advertise a large window size according to the dynamic right sizing.

(3) TCP in Mac OS X sometimes shows an intermittent type of retransmission which takes longer time than the others. In the experiment, it took several seconds to retransmit all the lost packets. The reason seems to be that the loss recovery based on the SACK option does not work well in Mac OS X. But, in other retransmissions, Mac OS TCP uses SACK based recovery, and so the clarification of Mac OS TCP behaviors is for further study.

REFERENCES

[1] 3GPP TS 23.009 version 7.0.0 Release 7, “Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Handover procedures,” ETSI, Mar. 2007.

[2] IEEE Std 802.17, “Part 17: Resilient packet ring (RPR) access method and physical layer specifications,” IEEE Standard Association, May 2011.

[3] Recommendation ITU-T G.8031/Y.1342, “Ethernet linear protection switching,” Telecommunication Standardization Sector of ITU, June 2011.

[4] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, “Host-to-Host Congestion Control for TCP,” IEEE Commun. Surveys Tutorials, vol. 12, no. 3, 3rd quarter 2010, pp. 304-340.

[5] F. Khafizov and M. Yavuz, “Running TCP over IS-2000,” Proc. ICC 2002, April 2002, pp. 3444-3448 vol. 5.

[6] E. Halepovic, Q. Wu, C. Williamson, and M. Ghaderi, “TCP over WiMAX: A Measurement Study,” Proc. IEEE MASCOTS 2008, Sept. 2008, pp. 1-10.

[7] J. Zhu and T, Bai, “Performance of Tahoe, Reno, and SACK TCP at Different Scenarios,” Proc. ICCT ’06, Nov. 2006, pp. 1-4.

[8] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment Options,” IETF RFC 2018, Oct. 1996.

[9] I. Rhee and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” SIGOPS Operating Systems Review, vol. 42, no. 5, July 2008, pp. 64-74.

[10] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCP Westwood: Bandwidth estimation for enhanced transport over wireless links,” Proc. ACM MOBICOM 2001, July 2001, pp. 287-297.

[11] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “A compound TCP approach for high-speed and long distance networks,” Procl IEEE INFOCOM 2006, April 2006, pp. 1-12.

[12] Wireshark Foundation, “WIRESHARK,” http://www.wireshark.org/

[13] M. Fisk and W. Feng, “Dynamic Right-Sizing in TCP,” Proc. Los Alamos Computer Science Institute Symposium, Oct. 2001.

[14] E, Blanton, M. Allman, L. Wang, I. Jarvinen, M. Kojo, and Y. Nishida, “A Conservative Loss Recovery Algorithm Based on Selective Acknowledgment (SACK) for TCP,” IETF RFC 6675, Aug. 2012.

[15] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, “An Extension to the Selective Acknowledgment (SACK) Option for TCP,” IETF RFC 2883, July 2000.

 ISBN: 978-1-61208-318-6

Optimizing Green Clouds through Legacy Network Infrastructure Management

Sergio Roberto Villarreal, Carlos Becker Westphall, Carla Merkle Westphall
Network and Management Laboratory - Post-Graduate Program in Computer Science
Federal University of Santa Catarina
Florianopolis, SC, Brazil
sergio@inf.ufsc.br, westphal@inf.ufsc.br, carlamw@inf.ufsc.br

Abstract — **The concepts proposed by Green IT have changed the priorities in the design of information systems and infrastructure, adding to traditional performance and cost requirements, the need for efficiency in energy consumption. The approach of Green Cloud Computing builds on the concepts of Green IT and Cloud in order to provide a flexible and efficient computing environment, but their strategies have not given much attention to the energy cost of the network equipment. While Green Networking has proposed principles and techniques that are being standardized and implemented in new networking equipment, there is a large amount of legacy equipment without these features in datacenters. In this paper, the basic principles pointed out in recent works for power management in legacy network equipment are presented, and a model for its use to optimize green cloud approach is proposed.**

Keywords - Green IT; Green Networking; Green Cloud Computing

I. INTRODUCTION

Traditionally, computer systems have been developed focusing on performance and cost, without much concern for their energy efficiency. However, with the advent of mobile devices, this feature has become a priority because of the need to increase the autonomy of the batteries.

Recently, the large concentration of equipment in data centers brought to light the costs of inefficient energy management in IT infrastructure, both in economic and environmental terms, which led to the adaptation and application of technologies and concepts developed for mobile computing in all IT equipment.

The term Green IT was coined to refer to this concern about the sustainability of IT and includes efforts to reduce its environmental impact during manufacturing, use and final disposal.

Cloud computing appears as an alternative to improve the efficiency of business processes, since from the point of view of the user, it decreases energy costs through the resources sharing and efficient and flexible sizing of the systems. Nevertheless, from the standpoint of the service provider, the actual cloud approach needs to be seen from the perspective of Green IT, in order to reduce energy consumption of the data center without affecting the system's performance. This approach is known as Green Cloud Computing [1].

Considering only IT equipment, the main cause of inefficiency in the data center is the low average utilization rate of the resources, usually less than 50%, mainly caused by the variability of the workload, which obliges to build the infrastructure to handle work peaks that rarely happen, but that would decrease the quality of service if the application was running on a server fully occupied [2].

The strategy used to deal with this situation is the workload consolidation that consists of allocating the entire workload in the minimum possible amount of physical resources to keep them with the highest possible occupancy, and put the unused physical resources in a state of low energy consumption. The challenge is how to handle unanticipated load peaks and the cost of activation of inactive resources. Virtualization, widely used in the Cloud approach, and the ability to migrate virtual machines have helped to implement this strategy with greater efficiency.

Strategies to improve efficiency in data centers have been based mainly on the servers, cooling systems and power supply systems, while the interconnection network, which represents an important proportion of consumption, has not received much attention, and the proposed algorithms for load consolidation of servers, usually disregard the consolidation of network traffic.

The concepts of Green IT, albeit late, have also achieved design and configuration of network equipment, leading to Green Networking, which has to deal with a central problem: the energy consumption of traditional network equipment is virtually independent of the traffic workload. The Green Networking has as main strategies proportional computing that applies to adjust both the equipment processing speed such as the links speed to the workload, and the traffic consolidation, which is implemented considering traffic patterns and turning off components not needed. According to Bianzino et al. [3], traditionally the networking system design has followed two principles diametrically opposed to the aims of Green Networking, over-sizing to support demand peaks and redundancy for the single purpose of assuming the task when other equipment fail. This fact makes Green Networking technically challenging, with the primary objective of introducing the concept of energy-aware design in networks without compromising performance or reliability.

While the techniques of Green Networking begin to be standardized and implemented in the new network equipment, a large amount of legacy equipment forms the

 ISBN: 978-1-61208-318-6

infrastructure of current data centers. In the works to be presented in the next section, it is shown that it is possible to manage properly these devices to make the network consumption roughly proportional to the workload.

Thereby, there is the need and the possibility to add, to the Green Cloud management systems, means of interaction with the data center network management system, to synchronize the workload consolidation and servers shutdown, with the needs of the network traffic consolidation.

Taking into account that the more efficient becomes the management of virtual machines and physical servers, the greater becomes the network participation in the total consumption of the data center, the need to include network equipment in green cloud model is reinforced.

In this article, the principles suggested in recent papers by several authors for power management in legacy network equipment are presented, and their application to optimize our approach of green cloud is proposed.

After this introduction, section 2 presents related works on which is based our proposal that is presented in section 3. Section 4 presents possible results of the application of the model and, finally, in section 5, concluding remarks and proposals for future work are stated.

II. RELATED WORK

Mahadevan et al. [4] present the results of an extensive research conducted to determine the consumption of a wide variety of network equipment in different conditions. The study was performed by measuring the consumption of equipment in production networks, which made it possible to characterize the energy expenditure depending on the configuration and use of the equipment, and determine a mathematical expression that allows calculating it with an accuracy of 2%. This expression determines that total consumption has a fixed component, which is the consumption with all ports off, and a variable component which depends on the number of active ports and the speed of each port.

Research has determined that the power consumed by the equipment is relatively independent of the traffic workload and the size of packets transmitted, and dependent on the amount of active ports and their speed. The energy saved is greater when the port speed is reduced from 1 Gbps to 100 Mbps, than from 100 Mbps to 10 Mbps.

This research also presents a table with the average time needed to achieve the operational state after the boot of each equipment category, and also demonstrates that the behavior of the current equipment is not proportional, as expected according to the proposals of the Green Networking, and therefore the application of traffic consolidation techniques have the potential to produce significant energy savings.

Mahadevam et al. [5], continuing the work presented in the preceding paragraphs, put the idea that the switches consumption should ideally be proportional to the traffic load, but as in legacy devices the reality is quite different,

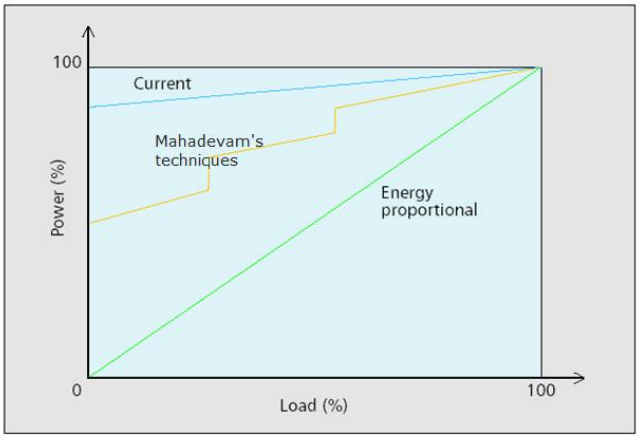


Figure 1 - Consumption in computer networks as a function of the workload [5].

they propose techniques to make the network consumption closer to the proportional behavior by the application of configurations available in all devices.

The results are illustrated in Figure 1, which shows the ideal behavior identified as "Energy Proportional" which corresponds to a network with fully "Energy Aware" equipment, the actual curve of the most of the today's networks where the consumption is virtually independent of load, labeled "Current", and finally the consumption curve obtained by applying the techniques they proposed, labeled “Mahadevam’s techniques”.

The recommended configurations are: slow down the ports with low use, turn off unused ports, turn off line cards that have all their ports off and turn off unused switches. The authors, through field measurements, have shown that it is possible to obtain savings of 35% in the consumption of a data center network with the application of these settings. Also, with the use of simulations, they have demonstrated that in ideal conditions savings of 74% are possible combining servers load consolidation and network traffic consolidation.

Werner [6] proposes a solution for the integrated control of servers and support systems for green cloud model based on the Theory of Organization (Organization Theory Model - OTM). This approach defines a model of allocation and distribution of virtual machines that were validated through simulations and showed to get up to 40% energy saving compared to traditional cloud model.

The proposed model determines when to turn off, resize or migrate virtual machines, and when to turn on or off physical machines based on the workload and the Service Level Agreement (SLA) requirements. The solution also envisages the shutdown of support systems. Figure 2 shows the architecture of the management system proposed, which is based on norms, roles, rules and beliefs.

Freitas [7] made extensions to the CloudSim simulator by CALHEIROS et al. [8], developed at the University of Melbourne, creating the necessary classes to support the Organization Theory Model, presented in the previous

 ISBN: 978-1-61208-318-6

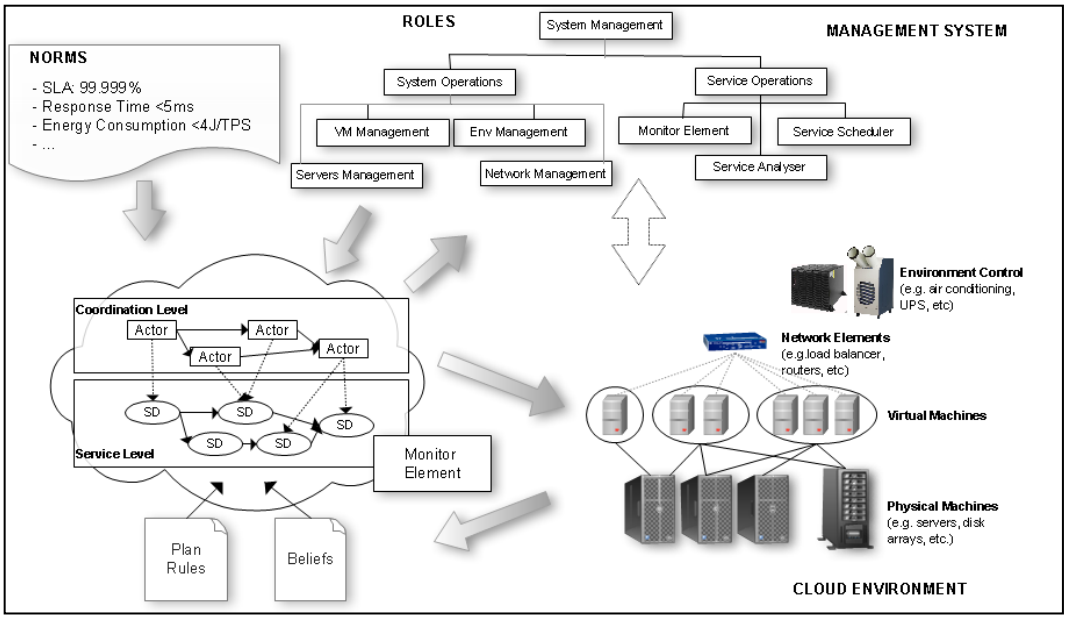


Figure 2 – Green Cloud management system based on OTM [6].

paragraphs, which allowed to calculate the energy savings and SLA violations in various scenarios.

In the next section, a proposal to include the management of legacy network devices in Organization Theory Model and the rules and beliefs for the proper functioning of the model based on the findings of the works described above are presented. The rules and equations required to include this extension in CloudSim simulations are also presented and validated through a study case.

III. PROPOSAL FOR DATA CENTER NETWORK MANAGEMENT IN GREEN CLOUD APPROACH

The proposal considers the network topology of a typical datacenter shown in Figure 3, where the switches are arranged in a hierarchy of three layers: core layer, aggregation layer and access or edge layer. In this configuration, there is redundancy in the connections between layers so that the failure of a device does not affect the connectivity.

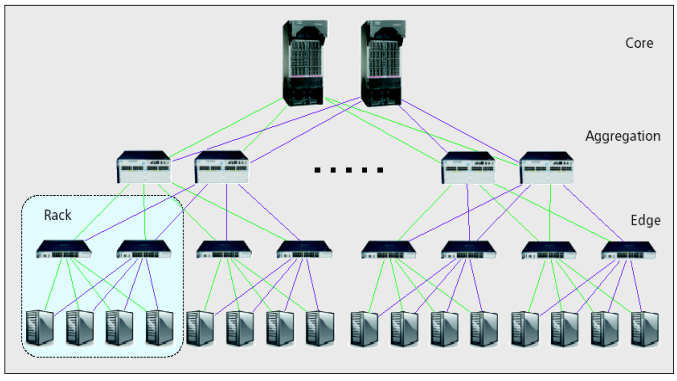


Figure 3- Typical network topology of a datacenter [5].

Consequently, we consider, in our model, that each rack accommodates forty 1U servers and two access layer switches. Each of these switches has 48 Gigabit Ethernet ports and two 10 Gigabit Ethernet uplink ports, and each server has two Gigabit Ethernet NICs each one connected to a different access switch.

We also consider that if there is only one rack, aggregation layer switches are not required, and up to 12 racks can be attended by 2 aggregation layer switches with twenty four 10 Gigabit Ethernet and two 10 Gigabit Ethernet or 40 Gigabit Ethernet uplinks, with no need for core switches.

Finally, the model assumes that, with more than 12 racks two core switches with a 24 ports module for every 144 racks will be required. The module's port speed may be 10 Gigabit Ethernet or 40 Gigabit Ethernet, according to the aggregation switches uplinks.

In traditional facilities, the implementation and management of this redundancy is done by the Spanning Tree Protocol and in most recent configurations by the Multichassis Links Aggregation Protocol (MC-LAG), which allows using redundant links simultaneously expanding its capacity, as described in [9].

A. *Extensions To The Organization Theory Model*

To include the management of legacy network equipment in the model proposed by WERNER et al [10], such that the network consumption becomes relatively proportional to the traffic workload and the energy savings contribute to the overall efficiency of the system, it is proposed to add the following elements to its architecture:

1) *Management Roles*

Add to the "System Operations" components the "Network Equipment Management" role, which acts as an

interface between the model and the network equipment being responsible for actions taken on these devices such as: enabling and disabling ports or equipment or change MC-LAG protocol settings.

The "Monitoring Management" role, responsible for collecting structure information and its understanding, should be augmented with elements for interaction with the network management system to provide data, from which decisions can be made about the port speed configuration, or turning on or off components and ports. These decisions will be guided by the rules and beliefs.

2) Planning Rules

These rules are used when decisions must be taken, and therefore, rules to configure the network equipment in accordance with the activation, deactivation and utilization of physical machines should be added.

To implement the settings pointed out in [5], already presented, the following rules are proposed:

- If a physical machine (PM) is switched off, the corresponding ports of access layer switches must be turned off.
- If the occupation of a PM is smaller than a preset value, network interfaces and corresponding access switches ports must be slowed down.
- If the aggregate bandwidth of the downlink ports of an access layer switch is smaller than a preset value, their uplink ports must have their speed reduced.
- If an access layer switch has all its ports off, it must be turned off.
- If an access layer switch is turned off, the corresponding ports of the aggregation layer switch must be turned off.
- If the aggregate bandwidth of the downlink ports of an aggregation layer switch is smaller than a preset value, their uplink ports must have their speed reduced.
- If an aggregation layer switch has all its ports off, it must be turned off.
- If an aggregation layer switch is turned off, the corresponding port of the core layer switch must be turned off.
- If a module of a core layer switch has all its ports off, it must be turned off.
- If a core layer switch has all its ports off, it must be turned off.
- All reversed rules must also be included.

The application of these rules does not affect the reliability of the network, since port and devices are only turned off when servers are turned off. The system performance will only be affected if the network equipment activation cost is bigger than the server activation cost.

For more efficiency in traffic consolidation, the model should consider the racks in virtual machines allocation and migration strategies, and rules that consolidate active physical machines in as fewer racks as possible are necessary.

3) Beliefs

They are a set of empirical knowledge used to improve decisions, and are linked to the used resources characteristics and to the type of services implemented in each specific case.

For each of the rules listed in the previous paragraph, a belief related to energy consumption should be stated. If we consider CHRISTENSEN et al. [11], examples include:

- Disconnecting a port on a switch access layer generates a saving of 500 mWh.
- Decreasing the speed of a port from 10 Gbps to 1 Gbps generates a saving of 4.5 Wh.

It will also be necessary to include beliefs about the time required for a deactivated port or device to become operational after the boot. These beliefs will be used to make decisions that must consider performance requirements.

B. Simulation Model

The typical datacenter network topology, rules and beliefs proposed form the basis for building a simulation model to validate different strategies and rules in specific settings and with different workloads. As already done in previous works by WERNER [6] and FREITAS [7], it is possible to expand the CloudSim [8] or work on some of its extensions as TeachCloud [12].

The simulator must create the network topology and calculate their initial consumption based on the amount of physical servers using the following rules:

- If the number of servers is smaller than 40, the topology will have only two access layer switches interconnected by their uplink ports. Turn off unused ports.
- If the number of servers is greater than 40 and smaller than 480 (12 Racks), put two access layer switches for every 40 servers or fraction and two aggregation layer switches interconnected by their uplink ports. Turn off unused ports of both layers switches.
- If the number of servers is greater than 480, apply the previous rule for each group of 480 servers or fraction, add two core layer switches and put on each switch a 24 ports module for each 5,760 servers (144 racks) or fraction. Turn off unused port.

The equation to calculate the consumption of the switches and modules is:

$$\text{Power (W)} = \text{BP} + \text{no. P 10Giga} \times 5 + \text{no. P Giga} \times 0{,}5 + \text{no. P Fast} \times 0{,}3 \quad (1)$$

In this expression, the power in Watts is calculated by summing the base power (BP), which is a fixed value specific to each device, and the consumption of every active port at each speed, which is the variable component. The consumption of each type of port is specific to each device, but the proposed values are the average values according to the works already cited.

 ISBN: 978-1-61208-318-6

In (1), if the switch is modular, the base power of the chassis must be added.

During the simulation, when servers are connected or disconnected, the simulator must apply the network management rules by turning on or off the corresponding ports or configuring its speed, and update the calculation of the total consumption of the network.

In order to analyze the system performance and SLA violations, the model must know the time needed to put into operation each type of equipment, and at the moment of the server's activation, compare the uptime of the server with the uptime of the network equipment and use the greatest.

IV. CASE STUDY

To validate the model and the potential of the proposal, it was applied to a hypothetical case of a cloud with 200 physical servers, creating the topology, calculating its initial consumption without network equipment management and illustrating two possible situations in the operation of the system. It was considered for this scenario that the base power is 60 W for access layer switches and 140 W for aggregation layer switches.

Applying the rule to calculate the topology, it is determined that it comprises 5 racks housing a cluster of 40 servers each and, therefore, there will be 10 access layer switches with 40 Gigabit Ethernet ports and two 10 Gigabit Ethernet empowered ports, and two aggregation layer switches with 12 connected ports each, 10 ports for access layer switches and two ports for uplink interconnection between them.

A. Scenario 1: All network equipment with all its ports connected

The consumption of the network will be:

$$\text{Access layer switches} = 10 \times (60 + 2\times5 + 48\times0{,}5) = 940 \text{ W} \quad (2)$$

$$\text{Aggregation layer switches} = 2 \times (140 + 24\times5) = 520 \text{ W} \quad (3)$$

$$\text{Total network consumption} = 1.460 \text{ W} \quad (4)$$

B. Scenario 2: Initial configuration with unused ports off

The consumption of the network will be:

$$\text{Access layer switches} = 10 \times (60 + 2\times5 + 40\times0{,}5) = 900 \text{ W} \quad (5)$$

$$\text{Aggregation layer switches} = 2 \times (140 + 12\times5) = 400 \text{ W} \quad (6)$$

$$\text{Total network consumption} = 1.300 \text{ W} \quad (7)$$

In this scenario, it is observed that only by the proper initial configuration of the network it is possible to get a power save of approximately 11%.

C. Scenario 3: 90 active servers, workload consolidated in the first three racks and network configuration rules applied.

In this situation, according to the rules, there are 4 access layer switches working in initial conditions (8), two access layer switches working with twelve 1 Gbps ports, 10 for servers and 2 uplink ports with its speed reduced (9), and 2 aggregation layer switches with four 1 Gbps ports and two 10 Gbps ports (10), and the network consumption will be:

$$\text{Access layer switches 1} = 4 \times (60 + 2\times5 + 40\times0{,}5) = 360 \text{ W} \quad (8)$$

$$\text{Access layer switches 2} = 2 \times (60 + 12\times0{,}5) = 132 \text{ W} \quad (9)$$

$$\text{Aggregation switches} = 2 \times (140 + 4\times5 + 2\times0{,}5) = 322 \text{ W} \quad (10)$$

$$\text{Total network consumption} = 814 \text{ W} \quad (11)$$

In this scenario, there is a power saving of approximately 45% in network consumption.

V. CONCLUSIONS

In this paper, basic concepts related to Green IT were first presented, i.e., Green Cloud and Green Networking, demonstrating the need of considering the network equipment in strategies designed to make data centers more efficient, since the network represents a significant percentage of total consumption, and this participation will be more expressive when the other components become more efficient.

Afterwards, in the related work section, a green cloud management model called Organization Theory Model (OTM) was presented, as well as network equipment management principles that, when properly applied, make the behavior of the total consumption of the network approximately proportional to the traffic load, even when legacy energy-agnostic equipment are used in. The proposal was to extend the OTM to manage the network traffic consolidation according to these management principles.

Then, the elements that must be added to the architecture of the OTM were described, including the rules and beliefs required for the correct network configuration according to the load consolidation on servers.

It was also proposed a model to determine the data center network topology based on the number of physical servers, the rules to manage and set the network devices according to the servers' state changes, and equations to calculate the switches consumption and the total network consumption. This model is the basis to create a simulator and perform simulations to test the viability and the impact of the proposal application in different configurations, with different performance requirements and with different rules and beliefs.

The model was validated by its application in a case study, which allowed verifying that equations and rules are correct and enough to create the topology and to calculate the consumption of the network in each step of the simulation, as

 ISBN: 978-1-61208-318-6

well as highlight the possible effects of the application of the proposal.

It was also demonstrated, that in the described scenario it is possible to get a power saving of approximately 11% only by the proper initial configuration of the network and without any compromise of the performance. In a hypothetical situation of low utilization as described in scenario 3, a power saving of approximately 45% through proper workload consolidation is possible. It was thus demonstrated the possibility and desirability of extending the green cloud management model as proposed.

It is important to consider that the impact of applying the model is maximum in legacy energy-agnostic equipment, and will be smaller as the equipment becomes more energy-aware by applying the resources of the Green Networking as described in [13], but its application will be still convenient.

As future research, it is proposed to continue this work by developing the necessary extensions to CloudSim to implement the model, and perform experiments to determine the most effective rules and virtual machine allocation policies, and the actual contribution of the model in scenarios with different configurations, real workloads and taking into account possible violations to the SLA.

To evaluate the applicability of the model, it is also proposed to determine, through simulation, how many times a day a port or a device is turned on and off in real scenarios, and its possible impact in equipment failure rate.

Finally, since system performance may be affected if the network devices activation cost is bigger than the server activation cost, it is also suggested to study the proper network configuration and technologies to avoid this situation, with special consideration to protocols that manage the links redundancy and aggregation, like the Spanning Tree Protocol, MC-LAG, and other new networking standards for data centers.

REFERENCES

[1] C. Westphall and S. Villarreal, "Principles and trends in Green Cloud Computing ", Revista Eletrônica de Sistemas de Informação, v. 12, n. 1, pp. 1-19, January 2013, doi: 10.5329/RESI.2013.1201007.

[2] A. Beloglazov, R. Buyya, Y.C. Lee, and A. Zomaya, "A taxonomy and Survey of Energy-efficient Datacenters and Cloud Computing". Advances in Computers, vol 82, pp. 47-111, Elsevier, November 2011, doi: 10.1016/B978-0-12-385512-1.00003-7.

[3] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of Green Networking research". IEEE Communications Surveys and Tutorials, vol 14, pp. 3-20, February 2012, doi: 10.1109/SURV.2011.113010.00106

[4] P. Mahadevan, P. Sharma, S. Banerjee and P. Ranganathan, A "Power Benchmarking Framework for Network Devices". Proc. 8th International IFIP-TC 6 Networking Conference, Springer Berlin Heidelberg, November 2009, pp. 795-808, doi: 10.1007/978-3-642-01399-7_62

[5] P. Mahadevan, S. Banerjee, P. Sharma, A. Shah, and P. Ranganathan, "On energy efficiency for enterprise and data center networks". IEEE Communication Magazine. vol. 49 pp. 94-100. August 2011. 10.1109/MCOM.2011.5978421

[6] J. Werner, "A virtual machines allocation approach in green cloud computing environments". dissertation: Post-Graduate Program in Computer ScienceFederal University of Santa Catarina, 2011.

[7] R. Freitas, "Efficient energy use for cloud computing through simulations". Monograph: Post-Graduate Program in Computer Science Federal University of Santa Catarina, 2011.

[8] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resources Provisioning Algorithms". SPE Wiley Press, vol. 41, pp. 23-50, January 2011.

[9] C. Sher De Cusatis, A. Carranza, and C. Decusatis, "Comunication within clouds: open standards and proprietary protocols for data center networking", IEEE Communication Magazine. Vol. 50, pp. 26-33, September 2012. doi: 10.1109/MCOM.2012.6295708

[10] J. Wener, G. Geronimo, C. Westphall, F. Koch, and R. Freitas, "Simulator improvements to validate the green cloud computing approach", LANOMS, October 2011, pp. 1-8, doi: 10.1109/LANOMS.2011.6102263

[11] K. Christensen, P. Reviriego, B. Nordman, M. Mostowfi, and J. Maestro, "IEEE 802.3az: The road to Energy Efficient Ethernet", IEEE Comunication Magazine, vol 48, pp. 50-56, November 2010. doi: 10.1109/MCOM.2010.5621967.

[12] Y. Jararweh, M. Kharbutli, and M. Alsaleh, "TeachCloud: A Cloud Computing Educational Toolkit". International Journal of Cloud Computing (IJCC), Vol. 2, No. 2/3, February 2013, pp. 237-257, doi:10.1504/IJCC.2013.055269.

[13] D-LINK. Green Technologies. Taipei: D-LINK, 2011, available at: http://www.dlinkgreen.com/energyefficiency.asp. Accessed on 13 June 2013.

 ISBN: 978-1-61208-318-6

A Spectrum Sharing Method based on Adaptive Threshold Management between Non-cooperative WiMAX/WiFi Providers

Yukika Maruyama, Kazuhiko Kinoshita,
and Koso Murakami
Graduate School of Information Science and Technology,
Osaka University
Suita, Osaka 565–0871, Japan
Email: {maruyama.yukika, kazuhiko,
murakami}@ist.osaka-u.ac.jp

Keita Kawano
Center for Information
Technology and Management,
Okayama University
Okayama, Okayama 700–8530, Japan
Email: keita@cc.okayama-u.ac.jp

***Abstract*—In recent years, the number of portable mobile terminal users is increasing with improvement of the wireless communication environment. In addition, mobile multimedia services also bring the increase of wireless traffic and lack of spectrum resources. As a technique for efficient spectrum use, spectrum sharing receives much attention. It dynamically assigns a spectrum channel from a larger area network such as WiMAX to smaller area networks such as WiFi-based wireless LANs and achieves to increase the total wireless communication capacity of this area. In a case where all providers are cooperative, the spectrum sharing can be realized easily and improve user throughput in average. In another case where providers are not cooperative, some spectrum trading methods based on auction theory had been proposed. Although such an existing method assumes that providers can estimate their gain and loss caused by spectrum sharing, however, it is very difficult in fact since they need to model the users' behavior exactly. In this paper, we propose a spectrum sharing method that works properly between non-cooperative WiMAX/WiFi providers. It is achieved by simple and adaptive parameter management. Finally, we confirm the effectiveness of the proposed method by simulation experiments.**

Keywords- WiMAX/WiFi Integrated Network; Spectrum Sharing; Spectrum Assignment; Cognitive Radio

I. INTRODUCTION

With advances of wireless transmission technology (e.g., WiMAX [1], [2], WiFi [3] and Cellular), mobile communication environment is greatly improved. People can get multimedia services via these networks and the demand will grow as much as in wired networks.

On the other hand, the available radio spectrum resources for a particular wireless systems are getting scarcer. Since radio spectrum is statically allocated to licensed wireless providers, some frequency bands are unused in any time and location. In order to improve the wireless spectrum utilization, effective integration of multiple wireless networks is required.

Utilizing the cognitive radio technology[4], wireless systems can share wireless spectrum in heterogeneous networks. With spectrum sharing, it was confirmed that the frequency usage is improved in WiMAX/WiFi integrated network [5]. However, the existing method assumed that the WiMAX and WiFi providers cooperated to improve mean user throughput. Therefore, in the case where the providers do not cooperate and pursue only their own interests, this method might not work properly.

In such a case, spectrum trading methods had been proposed [6], in which spectrum bands are bought and sold among providers. While most of these works assumes that providers can estimate their gain and loss caused by spectrum trading, it is very difficult in fact since users' behavior has to be modeled exactly.

In this paper, we propose a spectrum sharing method that works properly even between non-cooperative WiMAX/WiFi providers by introducing a threshold to assign an additional channel from a WiMAX base station (BS) to WiFi access points (APs). It is necessary to adapt the threshold of spectrum sharing to the environment. Therefore, we also propose an adaptive threshold management, which is a learning algorithm for the threshold to match the spectrum demand.

The rest of the paper is organized as follows. In Section 2, we introduce spectrum sharing technology and some existing methods. In Section 3, we elaborate our proposed method. Its performance is evaluated by simulation experiments in Section 4. Finally, Section 5 presents some conclusions and indicates future work.

II. SPECTRUM SHARING

A. Integrated Network

Currently, most of wireless systems are independently designed. The integration of those independent wireless systems, however, is able to provide wireless users a seamless access. Therefore, in recent years, integrated network such as the Cellular/WiFi integrated network [7], [8] have been researched actively. As a typical heterogeneous wireless integrated network, we focus on integration of WiMAX and WiFi. The former system whose coverage area is several kilometers wide can achieve Quality of Service (QoS). The latter system can cover only several hundred meters, although, spread widely.

 ISBN: 978-1-61208-318-6

As shown in Fig. 1, in the integrated network, mobile users are connected to the best wireless systems, which are chosen in terms of user (e.g., application and mobility) and system (e.g., traffic congestion). Therefore, users can have better communications and systems also achieve load balancing [9].

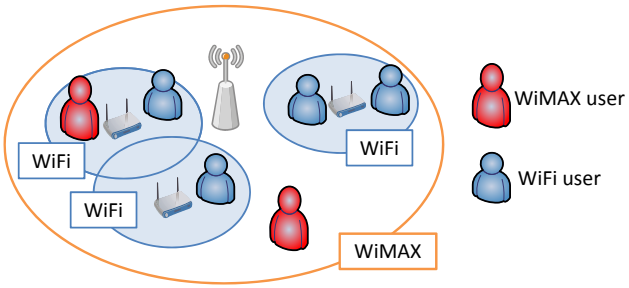


Figure 1: AP selection strategy

B. Spectrum Sharing Model

Spectrum sharing technology has emerged to improve the spectrum utilization in wireless networks. With the aid of cognitive radio, a spectrum owner (or primary system) shares their licensed spectrum for secondary systems, which has no priority to the band [10]. Since two or more secondary systems can use the same spectrum when they are not adjacent, the total wireless communication capacity increases. In the case of WiMAX/WiFi integrated network, spectrum channels of the WiMAX system are temporarily assigned to WiFi APs and it improves the whole network capacity[11].

As shown in Fig. 2, a centralized control server named spectrum manager controls the spectrum assignment and collects necessary information for the assignment from a WiMAX BS and WiFi APs inside the WiMAX BS service area [12]. Also, the server searches for the optimal assignment pattern of WiFi AP.

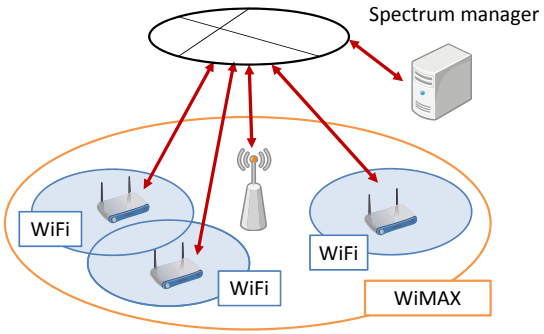


Figure 2: spectrum manager

C. Existing Spectrum Sharing Method

The protocol design for spectrum sharing depends on the relation between WiMAX and WiFi providers (i.e., cooperative or non-cooperative). In a cooperative environment, providers cooperate with each other to improve the average user throughput. On the other hand, non-cooperative providers pursue their own profit.

According to [5], spectrum sharing in cooperative network improves the overall average throughput. In this method, the number of users who connect to the system is used as the evaluation value of Genetic Algorithm (GA). By using GA under the constraint to disallow to assign the same spectrum to adjacent WiFi APs, it is possible to assign channel without interference among selected APs. However, this spectrum assignment method works properly only for a cooperative situation, for example, the same provider owns both WiMAX BS and WiFi APs.

In a non-cooperative environment, spectrum sharing may degrade the WiMAX system throughput due to the decrease of the available WiMAX spectrum. Therefore, spectrum trading method has been proposed. It adopts money trading as a motivation for spectrum sharing [6].

Spectrum trading method based on auction theory is one of the major methods to share frequency band in non-cooperative providers [13], [14]. In this method, each WiFi provider bids for an additional channel. By considering these offers, the WiMAX provider selects an assignment pattern that maximizes the WiMAX provider's revenue. This enables the WiMAX provider to obtain additional profit by lending channels and the WiFi providers to increase their effective bandwidth and user throughput. According to the profit which systems will get, providers decide to sell or buy channels. However, it is difficult to guess the right profit due to the difficulty to model exact user behavior, e.g., users may dynamically change to connect to a more comfortable system[14].

III. Proposed Method

A. Concept

We propose a spectrum sharing method based on the auction system with adaptive threshold management between non-cooperative WiMAX/WiFi providers. The proposed method relies on a simple management scheme by utilizing a threshold parameter which WiMAX provider sets as a minimum price per channel. On the other hand, the threshold control is important for an effective spectrum sharing. Therefore, WiMAX provider needs to learn and adapt the threshold parameter to the environment. For this purpose, we also propose a learning algorithm to decide an appropriate threshold. Note that this algorithm does NOT depend on the users' behavior modeling.

B. Network Model

We consider a heterogeneous network, that consists of one WiMAX BS and multiple WiFi APs contained in the BS covered area. Suppose the WiFi APi provides throughput T_i per a contacted user , for $i \in W_{all}$ (a set of all WiFi AP).

On a condition to leave at least one channel for BS, at most N-1 channels of WiMAX are provided to APs, where N is the number of channels primarily allocated to BS.

C. Threshold of Sharing Channel

Each WiFi AP submits a price for an additional channel based on the demand of the spectrum which can be described by provided throughput to each user. When an AP has larger number of connected users, each user receives less throughput. Therefore, price per channel U_i offered by WiFi APi is formulated as

$$U_i = \frac{x}{T_i}. \quad (1)$$

 ISBN: 978-1-61208-318-6

Since we suppose all APs have basically the same strategy to get an additional channel, x is a constant value.

Meanwhile, WiMAX provider sets the minimum price per channel y. The condition that WiMAX gives its spectrum resource to WiFi is described as

$$\sum_{i \in W_{GA}} U_i > y, \tag{2}$$

where W_{GA} is a set of WiFi APs receiving an additional channel. Though the number of available spectrum channels of the WiMAX BS decreases, this enables the WiMAX provider to obtain additional profit more than y by lending channels.

Since x is constant, x and y can put into one parameter M_{th} (WiMAX threshold) as

$$\sum_{i \in W_{GA}} \frac{1}{T_i} > \frac{y}{x} = M_{th}. \tag{3}$$

D. Channel Assignment

As shown in Fig. 3, a spectrum channel can be assigned to two or more APs which are not adjacent to each other. When a channel is assigned to an AP, the AP can use twice as much bandwidth. Thus, the spectrum demand of an AP which is assigned one or more channels will decrease and the AP will submit lower price for another additional channel. Therefore, WiFi AP changes the price of one channel by the number of the channels it receives.

The target APs for assignment and the number of assigned channels are decided according to the following steps.

1) WiFi APi decides the payment U_i.
2) WiMAX provider selects the assignment pattern that maximizes the sum of payments offered by WiFi APs.
3) If the revenue of lending the channels exceeds y, WiMAX provider performs the channel assignment.
4) Repeat Steps 1 to 3 until N-1 channels are lent or y exceeds the revenue from the target WiFi APs.

Note that, the condition of providing spectrum in Step 3 can be described by (2).

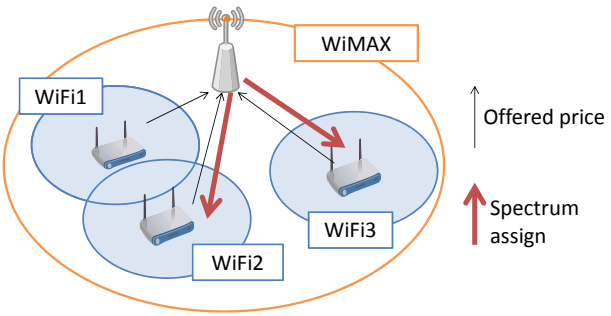


Figure 3: Channel assignment

E. Adaptive Threshold Management

When the threshold of sharing spectrum M_{th} is too small, a large part of the bandwidth is sold at a low price. On the other hand, frequency is not shared when M_{th} is too big. Therefore, it is necessary to set M_{th} to an adequate value in order to share spectrum effectively. Therefore, we propose a learning algorithm to find the appropriate M_{th} value dynamically.

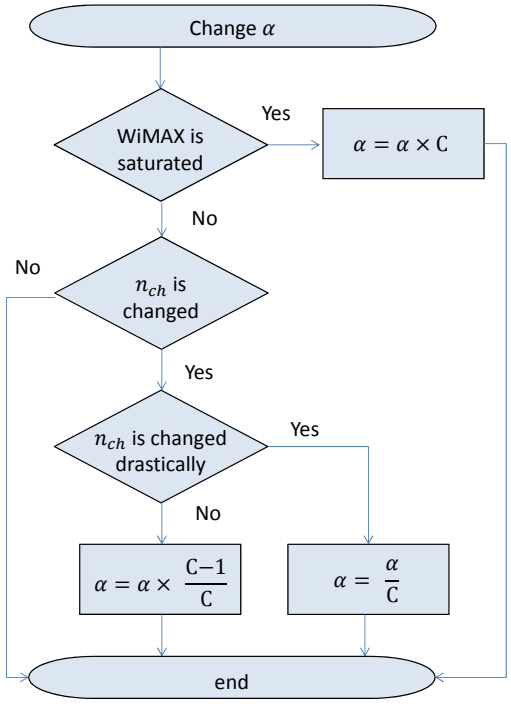


Figure 4: Flowchart of management α

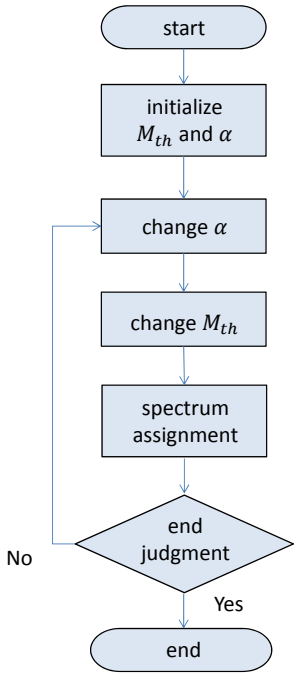


Figure 5: Flowchart of management M_{th}

According to the collected data of M_{th} and average users' throughput T_{all}, α is added or subtracted to M_{th}, where α is a variable. User throughput is given as

$$T_{usr} = \frac{T_{ch} \times C}{U}, \tag{4}$$

where T_{ch} is the maximum system throughput per one channel, C is the number of available channels for the connected system and U is the number of users connected to the system. M_{th} is varied by α to improve T_{all}. Also, α represents the sensitivity of M_{th}. We define N_{ch} as the number of assigned channels. Larger α leads to a drastic change of N_{ch}, so that M_{th} can be an adequate value quickly. In other words, however, M_{th} may be unstable. In the contrary, smaller α leads to a fine changes and more stable M_{th}, but results in slower adaptation time to reach an adequate M_{th} value. Therefore, we need to adjust α value as well. We propose an algorithm to adjust α value as shown in Fig. 4.

Specifically, when the number of users who connect to WiMAX keeps growing, α is set larger. In the case where the number of assigned channels changes greatly, such as zero to N-1, α is set smaller.

Fig. 5 shows the flowchart of the threshold management.

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed method to show its effectiveness by simulation experiments.

A. Simulation Model

As a network model, which is shown in Fig. 6, we set one WiMAX BS including $10 \times 10 = 100$ cells where randomly selected 50 cells had WiFi AP. APs in adjacent cells interfere with each other.

The spectrum bandwidth of the WiMAX was set to 100[MHz] and each WiFi AP was allocated in units of 20[MHz]. Each WiFi AP could use the one channel of 20[MHz] except any channels assigned from WiMAX. In addition, WiMAX BS was assumed to provide 20[Mbps] per 10[MHz] according to the evaluation in WiMAX Forum [15], and WiFi AP supported 17.5[Mbps] per channel according to our preliminary experiments using ns-2 [16]. We assumed that an AP could use any additional channels with no delay [17].

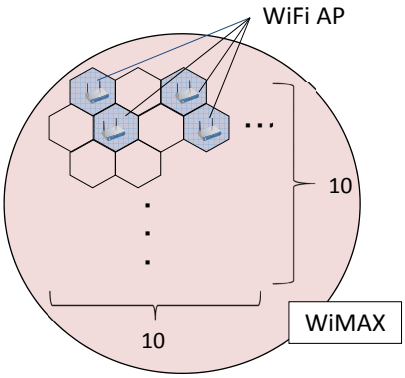


Figure 6: Network model

When a new user arrives at a cell with a WiFi AP, he/she connects to the WiFi and downloads a file. Otherwise, the user connects to the WiMAX. In addition, users were staying in the arrival cell until the end of downloading. Calls occurred following a Poisson arrival process.

We choose two methods for comparison. One is the existing spectrum assignment method [5]: described in section 2.3, this method allocates a spectrum to improve the overall mean throughput in a cooperative network. The existing method is a kind of ideal method in non-cooperative network. The other is the method that does not share any spectrum. As a performance measure, we observed the average throughput.

Moreover, to evaluate the performance of the adaptive M_{th} management, we adopt another compared method called $fixed\ method$ that uses a fixed M_{th}. The fixed M_{th} can be found by several trials and maximizes the average throughput.

The parameters we set are summarized in Table 1. Generally, because WiFi APs are set up in places where people gather (e.g., cafe and office), in order to emulate those environment, we differentiate the arrival rate in a cell with and without WiFi AP.

For M_{th} management, we set M_{th} and α to 0.1 as the initial value. Note that, we found that they are not so sensitive by preliminary experiments.

TABLE I: PARAMETERS SETUPS

interval time of spectrum assignment	300 [sec]
arrival rate in a cell with WiFi AP	λ [1/sec]
arrival rate in a cell without AP	0.1λ [1/sec]
file size	10 [MB]
traffic	best effort
initial M_{th}	0.1
initial α	0.1

B. Simulation Results

The number of assigned channels from WiMAX BS to WiFi APs with fixed M_{th} is shown in Fig. 7. For a smaller value of M_{th}, more and more spectrum channels are shared since WiFi AP can obtain the spectrum with lower cost. In contrast, for a larger M_{th}, it is difficult for APs to get channels. Therefore, M_{th} can control the number of the shared channels WiMAX and WiFi APs.

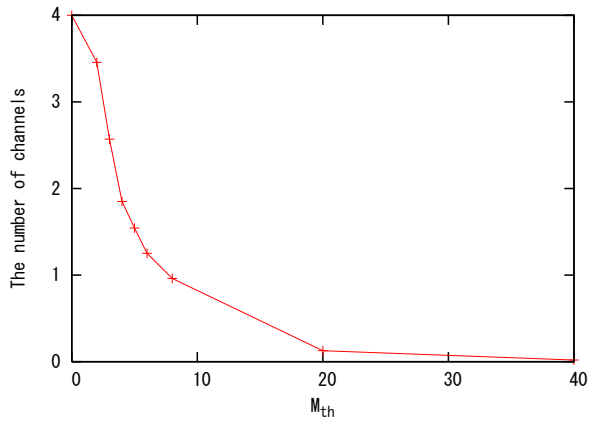


Figure 7: Mean number of assigned channel with variable M_{th}: $\lambda = 0.2$

In Fig. 8 under the condition where λ is fixed, M_{th} converges due to the adaptive M_{th} management. When λ is low, M_{th} is also low since the demand of the spectrum channels from the WiFi APs are low. As the price that the WiFi AP submits increase, M_{th} converge to a higher point to lent a channel at a high price.

Fig. 9 indicates that the existing method and the proposed method improve the average throughput against non-sharing method, since the capacity of the system increased by assigning channels from WiMAX BS to several WiFi APs. For a smaller λ, the average throughput of the proposed method is higher than that of the existing method since we introduce a learning algorithm and obtain feedback of the spectrum assignment interval time. On the other hand, for a larger λ, the average throughput of the proposal method is lower because the bigger λ, results in the higher M_{th}, due to this, the algorithm tries to converges to a smaller value resulting in a longer transient state.

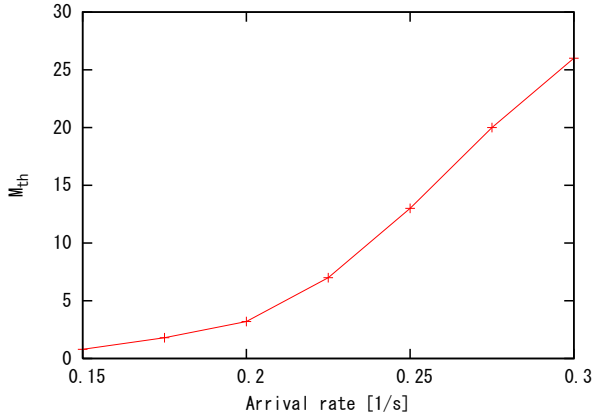


Figure 8: Converged M_{th} with variable λ

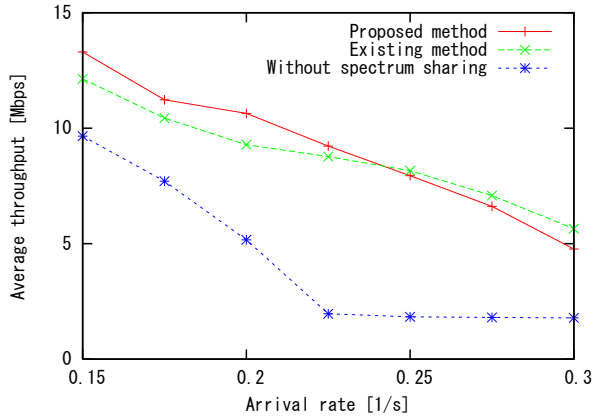


Figure 9: Average throughput with variable λ

The example of transient state of adaptive M_{th} management is shown in Fig. 10. In the transient state, which is soon after the start of spectrum sharing, users' throughput of the proposed method is lower since M_{th} is so small that many frequency bands of WiMAX BS are lent out. After the transient state passes, the overall throughput of the proposed method is as high as that of the fixed method.

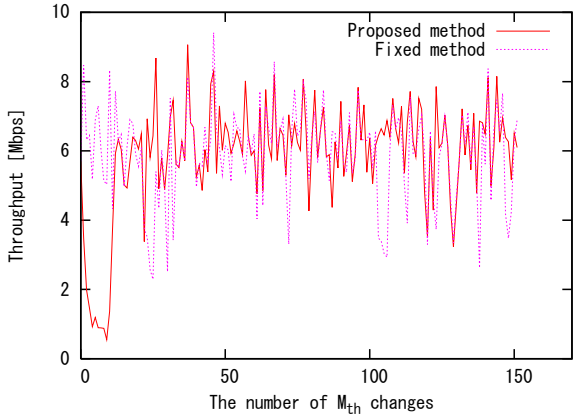


Figure 10: Overall throughput under M_{th} changes : $\lambda = 0.2$, fixed $M_{th} = 3.8$

The result shows that M_{th} can adapt to the environment where λ is fixed. Consequently, it is confirmed that the proposed method can assign spectrum well even if the WiMAX provider and WiFi providers are non-cooperative.

Moreover, according to [18], the mobile communication traffic in a day varies. It states that the traffic reach its peak at 23.00 and then decreases over the early morning. In addition, there is a three times difference in the maximum and the minimum traffic. To conform with this fact, we model the variation of λ as shown in Fig. 11 and observe the average of users' throughput over 30 days.

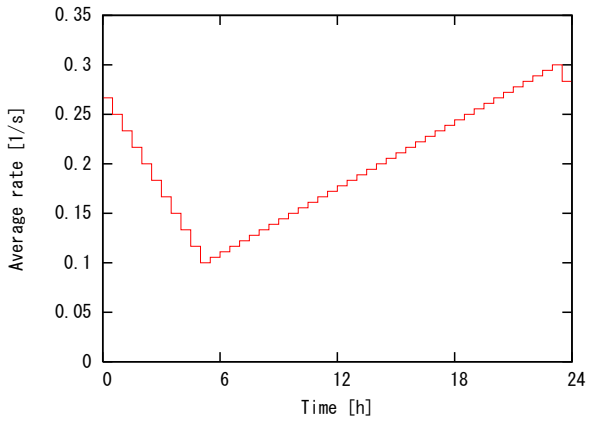


Figure 11: Time-varying λ

Figs. 12 and 13 show the change of M_{th} and the average throughput under time-varying λ. In the morning with a low traffic, M_{th} is low to share spectrum easily. In contrast, the heavier the mobile traffic, the higher M_{th} to refrain from loaning channels out cheaply.

Fig. 13 indicates that although proposed method includes a delay in comparison with the existing method, the users' throughput of the proposed method is much higher than that of the non-sharing method. Therefore, the results confirm that the proposal method can improve throughput in a practical situation.

 ISBN: 978-1-61208-318-6

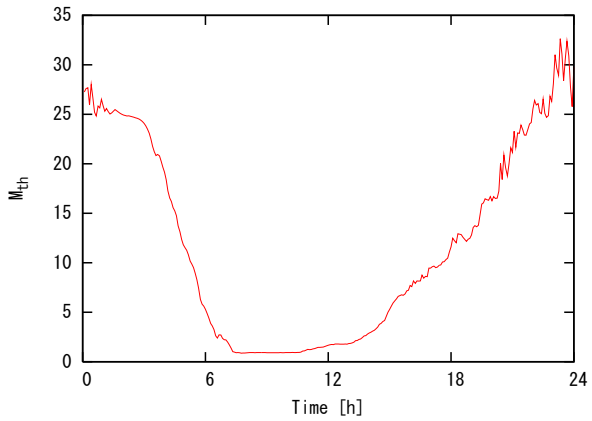


Figure 12: M_{th} with time-varying λ

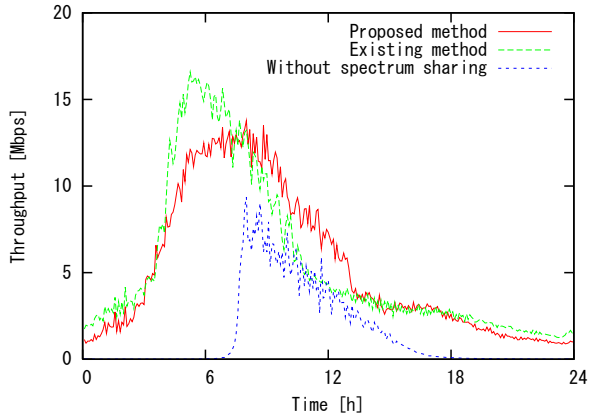


Figure 13: Overall throughput with time-varying λ

V. Conclusions

In this paper, we described the advances in wireless communication technologies and the lack of frequency resources. Next, we introduced integrated wireless network and spectrum sharing technology in heterogeneous integrated network. Different spectrum sharing methods and their problems were outlined. We proposed a spectrum sharing method using a minimum channel price threshold, which enabled WiMAX to control the spectrum sharing between non-cooperative WiMAX/WiFi providers. In addition, adaptive parameter management was proposed. Finally, we showed that the proposed method could assign spectrum efficiently and improve the user throughput by simulation experiments.

As a future work, we consider to improve the adaptation speed of the threshold and enhance the proposed method that supports multiple WiMAX BSs consideration.

Acknowledgements

This work is partly supported by Strategic Information and Communications R&D Promotion Programme (SCOPE), the Ministry of Public Management, Home Affairs, Posts and Telecommunications (MPHPT), Japan.

References

[1] "Air Interface for Fixed Broadband Wireless Access Systems," IEEE STD 802.16–2004, Oct. 2004.

[2] "Air Interface for Broadband Wireless Access Systems," IEEE STD 802.16m, Mar. 2011.

[3] IEEE 802.11, http://grouper.ieee.org/groups/802/11/ [Dec, 2012].

[4] I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, "A Survey on Spectrum Management in Cognitive Radio Networks," IEEE Communications Magazine, vol.46, Apr. 2008, pp.40–48.

[5] M. Nakagawa, K. Kawano, K. Kinoshita, and K. Murakami, "A Spectrum Assignment Method based on Genetic Algorithm in WiMAX/WiFi Integrated Network," ACM CoNEXT, Dec. 2009, pp.11–12.

[6] D. Niyato and E. Hossain, "Spectrum Trading in Cognitive Radio Networks: A Market-Equilibrium-Based Approach," IEEE Wireless Communications, vol.15, Dec. 2008, pp.71–80.

[7] W. Song, W. Zhuang, and Y. Cheng, "Load Balancing for Cellular/WLAN Integrated Networks," IEEE Network, vol.21, no.1, Jan.-Feb. 2007, pp.27–33.

[8] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When Cellular Meets WiFi in Wireless Small Cell Networks," IEEE Comunications Magazine, vol.51, June 2013, pp.44–50.

[9] S. Hanaoka, J. Yamamoto, and M. Yano, "Platform for Load Balancing and Throughput Enhancement with Cognitive Radio," IEICE Transactions on Communications, vol.E91-B, no.8, Aug. 2008, pp.2501–2508.

[10] J. M. Peha, "Approaches to Spectrum Sharing," IEEE Communications Magazine, vol.43, Feb. 2005, pp.10–12.

[11] D. Niyato and E. Hossain, "Wireless Broadband Access: Integration of WiMAX and WiFi: Optimal Pricing for Bandwidth Sharing," IEEE Communications Magazine, vol.45, no.5, May 2007, pp.140–146.

[12] K. Kinoshita, Y. Kanamori, K. Kawano, and K. Murakami, "A Dynamic Spectrum Assignment Method for Call Blocking Probability Reduction in WiFi/WiMAX Integrated Networks," IEICE Transactions on Communications, vol. E94-B, no.12, Dec. 2011, pp.3498-3504.

[13] J. Huang, R. A. Berry, and M. L. Honig, "Spectrum Sharing in Cognitive Radio Networks – An Auction based Approach," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol.40, June 2010, pp.587–596.

[14] H. Takemoto, K. Kawano, K. Kinoshita, and K. Murakami, "A Spectrum Sharing Method Considering Users' Behavior for Uncooperative WiFi/WiMAX Providers," ICN, Jan. 2013, pp.15–20.

[15] WiMAX Forum, http://www.wimaxforum.org/ [Dec, 2012].

[16] ns-2, The Network Simulator http://nsnam.isi.edu/nsnam/index.php/ [Dec, 2012].

[17] J. Xiao, R.Q. Hu, Y. Qian, L. Gong, and B. Wang, "Expanding LTE network spectrum with cognitive radios: From concept to implementation," IEEE Wireless Communications, vol.20, Apr. 2013, pp.12–19.

[18] Ministry of Internal Affairs and Communications (Japan), "Waga Kunino Ido Tsusin Traffic No Genzyo (Heisei 25Nendo 3Gatsu Bun)" (in Japanese) http://www.soumu.go.jp/johotsusintokei/field/ [Dec, 2012].

 ISBN: 978-1-61208-318-6

A Policy for Group Vertical Handover Attempts

Nivia Cruz Quental and Paulo André da S. Gonçalves
Centro de Informática – CIn
Universidade Federal de Pernambuco – UFPE
Recife, Brasil
Email: ncq@cin.ufpe.br, pasg@cin.ufpe.br

***Abstract*—The rising of heterogeneous networks brings vertical handover as an important topic in research. Current challenges include proposing efficient handover schemes or adapting classic existing schemes. In this paper, we propose a policy for Group Vertical Handover (GVHO) attempts. We evaluate our solution by modifying an existing GVHO scheme. Such scheme handles vertical group handovers based on a threshold that limits handover blocking probability. Although our study was made based on a specific scheme, the proposed solution is generic enough to be applied in other GVHO schemes. Results show that our solution reduces the handover latency in comparison to the original GVHO scheme studied while maintaining the handover blocking probability under a pre-defined threshold. In particular, we could reduce latency from 20% to 40% in the scenarios studied.**

Keywords-GVHO; handover; attempt; latency.

I. INTRODUCTION

Recently, the concept of handover (or handoff) has evolved to take into account the continuity of communication sessions even among different Radio Access Technologies (RATs) [1]. This replaces the classic concept of transferring an ongoing call or data session from one channel to another over networks using the same technology. The new definition is motivated by the recent popularity of devices such as tablets and smartphones, which are capable of supporting multiple link-layer technologies and handling different kinds of traffic. Inter-RATs handover is the main concern in Vertical Handover (VHO) studies [1]–[9] and issues such as the continuity of telephone calls and streaming sessions may also define requirements for handover decisions. Use cases in trains and on buses introduce new challenges. This leads us to look into the Group Handover (GHO) problem [10]. GHO takes place when two or more Mobile Nodes (MNs) intend to request handover at the same time to the same base station. During GHO, MNs are not necessarily aware of the presence of each other. Thus, GHO procedures must carry out load balancing. To achieve this, criteria such as energy saving, available bandwidth, and type of service may be considered [11]. Research related to Group Vertical Handover (GVHO) covers simultaneously issues from GHO and VHO [12]. GVHO brings the complexity of associating load balancing needs with the implications of choosing one technology instead of another. It must also handle legacy systems and individual handovers.

Among the handover phases of discovery, decision, and execution [9], the decision phase interests us the most. The decision process in GVHO is still an open issue and it may impact the GVHO overall performance, not only the decision algorithm itself, but the policy for GVHO attempts. GVHO research seeks to provide efficient decision techniques. Some of them are based on centralized entities [13], distributed algorithms [14], random delays [12], reinforcement learning [13], game theory [12], and optimization problems [11]. We give special attention to Lee *et al.* [11], since it addresses the latency reduction while considering load balancing, support to legacy networks, and handover blocking probability. Those issues are fundamental for advances in GVHO. The objective of Lee *et al.* [11] is to model GVHO decision as an optimization problem. Latency is minimized given the condition of maintaining the handover blocking probability under a pre-defined threshold. Although Lee *et al.* [11] present encouraging results, the scheme does not scale well. As the number of MNs grows, we have noticed a pronounced increase of latency.

We believe that if the handover scheme could control efficiently handover attempts, performance might be enhanced and the latency increase might be controlled as the number of MNs grows. In this paper, we propose a policy for handover attempts that is based on exponential backoff and uses information from the GVHO scheme itself. The proposed solution reduces average latency and eases the slope of the latency curve in comparison to results found in [11]. The main objective of this paper is to show the importance of choosing a proper policy for GVHO attempts. This paper is organized as follows: we present GVHO concepts in Section II. We present related work in Section III. We detail the scheme proposed in [11] in Section IV. We present the proposed policy for GVHO attempts in Section V. We present a comparative performance evaluation between the scheme with and without the proposed solution in Section VI. Finally, we highlight our conclusions in Section VII.

II. GROUP VERTICAL HANDOVER - GVHO

Recently, technological evolution has allowed the rising of cheaper gadgets supplied with multiple network interfaces. This fact has encouraged new research in mobility management considering brand-new use-cases. A remarkable challenge is to manage different connections taking place at the same time in public spaces with a diverse number of available technologies. The problem of a high number of users connecting simultaneously to the same base station supporting a different technology from their previous base station is studied in the Group Vertical Handover (GVHO) area of interest [1]. A GVHO

 ISBN: 978-1-61208-318-6

scenario is illustrated in Figure 1. Suppose an open event, like a music festival where users desire to communicate with friends, transmit multimedia data, and are constantly changing their location. There may be several available RATs and dozens of devices in communication sessions simultaneously.

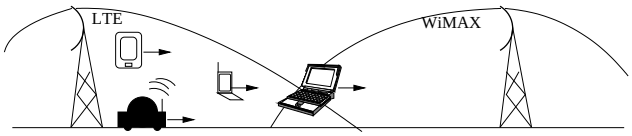


Figure 1. A GVHO scenario.

The proposal of efficient and effective handover procedures for mobility management including handover decisions and optimal resource allocation is a critical need for GVHO [1]. Proposals may involve the three handover phases [9]:

- *Discovery* - Service discovery and network information gathering. A specific criterion is adopted to determine if handover is necessary. It may be signal-to-noise ratio, transmission rate, Point of Attachment (PoA) load, battery consumption, etc.
- *Decision* - One network in a list of candidates is chosen, taking into consideration data collected in the earlier phase. This phase is the focus in this paper because the decision technique and the policy for handover attempts strongly impact the overall handover performance. Depending on the network technology, handover may be MN-initiated or network-initiated.
- *Execution* - Networks and MNs exchange control messages to make channel switching. This phase should minimize service interruption in order to appear imperceptible to the user. This phase is strongly media-dependent.

IEEE 802.21 standard [5] describes the Media Independent Handover Function (MIHF). MIHF intends to be a common mean over the link layer in order to allow different RATs to communicate with each other during handover. Each RAT must provide its own implementation of MIHF and map the MIHF messages to its media-dependent primitives. Thus, MIHF may offer information that can be used as discovery or decision parameters [11] [15]–[17].

There may be many different decision criteria for GVHO such as available bandwidth, expected QoS, or battery consumption. The type of service (voice or data) is a determinant factor for choosing the most suitable criterion for GVHO decision. Decisions made without network analysis and without considering the MNs in the neighborhood may bring disastrous performance results. Wrong handover decisions may cause MNs to choose the same PoA, overloading it, or to choose an inadequate network for the application in use. The main handover decision approaches found in GVHO research include:

- *Centralized entities* [13] - A relay station handles GVHO management, removing complexity from MNs. This approach also reduces the uncertainty level and ensures better performance than decentralized approaches. The main drawback is the lower fault tolerance.
- *Distributed algorithms* [14] - The decision algorithm makes use of well-known parallelism and synchronization techniques. Distributed algorithms are usually simple to understand. However, they are not built to adapt themselves to new scenarios.
- *Random delays* [12] - MNs attempt to handover after a random delay. This procedure minimizes simultaneous handover attempts and is considered a subtype of the distributed algorithm approach.
- *Reinforcement learning* [13] - It employs Artificial Intelligence (AI) techniques to make MNs learn about their surrounding environment as they make handover attempts. This approach does not require message exchange among users. However, learning algorithms can cause performance issues.
- *Game theory* [12] [13] - This approach maps handover scenarios in cooperative or non-cooperative games in which MNs are players interested in getting the best payoff as possible. The payoff may be a larger bandwidth, energy saving, or better security. Nash equilibrium is the desired stable state in which all MNs do not have anymore strategies to obtain better payoffs. The main advantage of this approach is the almost perfect match between a GVHO scenario and the Game Theory competitive models. On the other hand, it is not always possible to model additional parameters.
- *Mathematical optimization problems* [11] - Mathematical equations are used to describe the handover decision under predetermined conditions. Then, the problem is solved by finding the ideal value for the equation variables. This approach requires a more complex modelling and is more flexible than Game Theory-based models.

For any GVHO approach, the MN or the serving PoA may determine if it is possible to request handover in a certain time, or if it is preferable to postpone it, given the network conditions. Policies for handover attempts can influence handover performance, for better or for worse, depending upon the adopted solution.

III. Related Work

In [18], a relay station is used as a centralized entity to coordinate GVHO. The scenario studied is the movement of users in a train. Handover blocking and interruption probabilities are evaluated with the increase of the calls-per-minute ratio. The evaluation compares schemes with and without the relay station. The authors conclude that the proposed scheme reduces handover blocking and interruption probabilities. In this case, the relay station is responsible for executing the policy for handover attempts. The solution has limitations if co-existence with legacy systems is needed. This is due to the need of introducing a new infrastructure with special requirements.

 ISBN: 978-1-61208-318-6

Cai *et al.* propose three decentralized algorithms for GVHO in [12]. The first is a Nash equilibrium-based algorithm where the policy for handover attempts is based on the game strategy of each player. The second algorithm adopts random delays, thus using a simpler policy for handover attempts. The third algorithm is a more refined version of the previous one. It considers latency as a basis for delay calculations. Performance evaluations show that latency values under the three algorithms are similar. Handover blocking probability is not considered. Handover blocking probability is the probability of the MN having its handover request denied by the target network.

Niyato *et al.* propose a model for network selection that is based on evolutionary games [13]. The model consider two approaches: a central entity-based approach and a decentralized-based approach that uses a reinforcement learning model. In the first approach, the central entity controls handover attempts. In the second approach, MNs are allowed to infer the best period of time to request a handover. The fraction of MNs choosing the same PoA is the load-balancing metric adopted. They conclude that each approach has its advantages in accordance with the scenario. One drawback is not evaluating the impact of the approaches on latency.

Lei *et al.* [14] present three GVHO schemes. The first scheme schedules simultaneous attempts to random time periods. In the second scheme, MNs select PoAs using a predefined probability as a base. In this case, the policy of handover attempts consists in an immediate attempt. The last scheme requires the network to be responsible for the handover decision. Results show that the last approach is more efficient. However, it may be difficult to adapt it to legacy systems.

Lee *et al.* [11] propose a GVHO scheme, which is based on the solution of an optimization problem. The main objective is to minimize latency while limiting the handover blocking probability. Some factors make the scheme in [11] more promising than the other researches:

- it does not require the presence of a relay station.
- it may work together with legacy systems.
- it considers two of the main GVHO metrics: load balancing and latency.

Despite of presenting a promising GVHO scheme, the work in [11] lacks a good policy for handover attempts. It is based on a constant delay, which causes a negative impact on the overall GVHO performance as the number of MNs increases. We detail such scheme in Section IV.

IV. Reference GVHO scheme

Lee *et al.* [11] propose an optimization for the total handover latency L, considering the handover blocking probability as follows:

Minimize L

Subject to $P_{HoBlock}(t) \leq P_{HoBlockThreshold}$,

where $P_{HoBlock}(t)$ is the handover blocking probability in a time t and $P_{HoBlockThreshold}$ is the maximum acceptable value for the handover blocking probability. Latency is calculated as follows:

$$L = N_{HO}.\Delta t, \tag{1}$$

where N_{HO} is the total number of attempts until the MN requests the handover; Δt is the period of time between consecutive attempts. If the MN decides to request in the first attempt, total latency would be Δt. This is because in [11], execution time is also equal to Δt.

Equation (2) presents the calculation of $P_{HoBlock}(t)$. The value of $P_{HoBlock}(t)$ is dependent on the number of candidate networks, their available bandwidth, and the number of participating MNs in GVHO. In [11], it is considered that these values can be obtained by using IEEE 802.21 MIH (*Media Independent Handover*) queries and *ad hoc* communication.

$$P_{HoBlock}(t) = \sum_{k=1}^{K} \sum_{i=C_k(t)}^{M-1} \frac{(i+1-C_k(t)).(M-1)!}{(i+1)!.(M-1-i)!} \times ((P_{sel}^k)^{i+1}.(1-P_{sel}^k)^{M-1-i}), \tag{2}$$

where:

- M represents the number of participating MNs.
- K represents the number of candidate networks with overlapping areas.
- $C_k(t)$ is the available bandwidth in a time *t* for a network$_k$. The model considers that the available bandwidth is represented by an integer value. Each MN takes one unity for handover;
- P_{sel}^k : The probability of selecting network$_k$.

The Karush-Kuhn-Tucker (KKT) condition is used in optimization problems and it can be applied to (2) to determine the P_{sel}^k value. However, P_{sel}^k can be obtained by using (3), which is simpler than using KKT and induces minor changes in results.

$$P_{sel}^k(t) = C_k(t) / \sum_{k=1}^{K} C_k(t). \tag{3}$$

Now, we can find the $M_{optimal}(t)$ value that ensures the optimization problem condition. This value can be found by setting it initially to one, then increasing it by one unit while the $P_{HoBlock}(t)$ value is still less than or equal to $P_{HoBlockThreshold}$. The probability $P_{HO}(t)$ with which a MN can request handover is given by:

$$P_{HO}(t) = M_{optimal}(t)/M. \tag{4}$$

If the MN decides not to request the handover immediately, a new attempt will be made after a constant time interval. The MN requires the number of attempts necessary to have a well-succeeded handover with blocking probability less than or equal to $P_{HoBlockThreshold}$. Algorithm (1) summarizes this process and can also be found in [11]:

where:

- `Mtotal` is the total number of MNs in GVHO.
- `Mremaining` is a counter that checks for the algorithm end.
- `decision()` is a function that returns `true` with probability $P_{HO}(t)$.
- `LHOexec` is the handover execution time. It is equal to Δt.

 ISBN: 978-1-61208-318-6

Algorithm 1: Reference GVHO scheme

```
L = 0;
c_atts = 1;
Mtotal = number of GVHO participants;
Mremaining = Mtotal;
while Mremaining ≤ 0 do
    find Moptimal in function of (2);
    calculate PHO;
    if decision(PHO) then
        choose networkk depending on P^k_sel;
        NHO = c_atts;
        break ;
    else
        L += t_atts(c_atts);
        c_atts++;
    end
    Mremaining = Mremaining - Moptimal
end
L += LHOexec;
```

- `t_atts()` is a function to calculate the period of time between consecutive attempts. In [11], the return value of this function is always Δt.
- `c_atts` counts the number of attempts. When `decision()` is `true` in the first attempt, the total execution latency is L_{HOexec}.

Function `t_atts()` characterizes the policy for handover attempts. In [11], the return value of this function is constant and equals to the execution latency L_{HOexec}. We observe that the increase of latency is directly related with the number of attempts. Latency always grows by a constant factor because of `t_atts()`. We conclude that this policy of handover attempts does not take advantage of information provided by the scheme itself. Additionally, it causes a negative effect in the overall handover performance as the number of MN grows, as shown in [11].

V. The Proposed Policy for GVHO Attempts

In this section, we present a policy for GVHO attempts that aims at providing reduced handover latency for GVHO schemes like the one proposed in [11]. At the same time, we intend to reduce the slope of the latency curve as the number of MNs grows.

In order to enhance performance results, we propose to modify the `t_atts()` function in Algorithm (1). Our proposed solution is exponential backoff-based. It depends upon the `c_atts` counter and the duration of a reference slot time. It is a particular case of random delay. Exponential backoff algorithms have the particularity of keeping the probability of collision and the probability of transmission stable as the number of nodes which are sharing a medium grows [19]. Although our solution is motivated by the performance issues in [11], it is generic enough to be applied in other schemes. Equation (5) shows our modified version of `t_atts()`:

$$t\_atts(c\_atts) = \begin{cases} random[0..2^{c\_atts} - 1] \ . \ timeSlot \ , \\ \quad if \ c\_atts \leq LimBackFactor \\ random[0..2^{LimBackFactor} - 1] \ . \ timeSlot \ , \\ \quad otherwise \end{cases} \tag{5}$$

where $random$ picks a uniformily distributed number over the given interval; $LimBackFactor$ is the number of attempts that limits the range of values for $random$; and $timeSlot$ is the duration of a reference time slot, which depends on the target network. This information is obtained via MIH.

Total latency depends directly on the number of attempts, which varies with the return of `decision()`. The exponential backoff approach in `t_atts()` give to the MN an opportunity for a new handover attempt after a time interval shorter than Δt, or even immediately. When the MN chooses not to request handover, other MNs may request it, reducing concurrency during the next attempts. Thus, the total number of attempts reduces, decreasing total latency and easing the slope of the latency curve as the number of MNs grows.

VI. Performance Evaluation and Comparison

The metrics evaluated are the same as in [11]: latency and handover blocking probability, both *versus* the number of MNs. The majority of the parameters also follows the work in [11]. The value of Δt is set to *0.1s*. We study scenarios with different values for $P_{HoBlockThreshold}$: *0.02* and *0.05*. Telecordia (formerly Bellcore) [20] recommends a value of *0.01* as a QoS objective. However, typical values range around *0.02* [21] [22]. We consider the value of *0.05* for $P_{HoBlockThreshold}$ in order to observe the effects of choosing a less conservative probability. The number of MNs varies from 20 to 100. It differs from Lee *et al.* [11], where this number varies from 20 to 65. The characterization of heterogeneity in simulations presented by Lee *et al.* [11] is made through the use of different available bandwidths. The number of available PoAs is 3, considering the following scenarios:

Scenario 1 - All PoAs have 18 bandwidth units.

Scenario 2 - PoAs have 5, 13, and 18 bandwidth units, respectively.

Scenario 2 is only used in [11] for validating their simulator and in a situation of co-existing individual handover, which is out of the scope of this paper. Nevertheless, we include Scenario 2 in our evaluations. The $FatorLimBack$ parameter is set to *10*. This value is based on preliminary experiments. We consider that MNs are switching from an arbitrary network to an IEEE 802.11 area. The parameter $timeSlot$ is set to $9.10^{-6}s$, which is equivalent to the SIFS time slot in IEEE 802.11 standard. We have implemented the reference scheme and our solution in a discrete-event simulator, which was written in C++. The implementation of the reference scheme in our simulator was validated by the authors of [11]. We consider a group of MNs simultaneously entering a new coverage area and starting handover procedures defined by the GVHO scheme studied. We represent confidence intervals with 99% of confidence level. Confidence intervals appear imperceptible in Figures 2-5. It is important to point out that we are not interested in evaluating the decision algorithm itself, but the impact of our policy for GVHO attempts on performance.

 ISBN: 978-1-61208-318-6

Figure 2 shows results for handover blocking probability under Scenario 1. The probability increases as the number of MNs grows to 45 for threshold *0.02* and to 50 MNs for threshold *0.05*. Thereafter, the curves are stable. This happens because blocking probability is getting closer to the threshold defined in the optimization problem. Since blocking probability is directly related to the cell utilization [23], it is necessary to limit the number of MNs entering a new cell at the same time in order to maintain the blocking probability under the threshold. When the blocking probability reaches the threshold, the value of $M_{optimal}(t)$ that is calculated in function of (2) can not increase anymore. This leads the remaining MNs to wait for another handover attempt. Thus, the stabilization of the blocking probability curve as the number of MN grows always implies the increase of the average latency. It is important to notice that the curves with and without our solution are similar because the optimization problem conditions are still the same. It means that the application of the proposed solution does not cause damages to the handover blocking probability, despite of the shorter time between attempts.

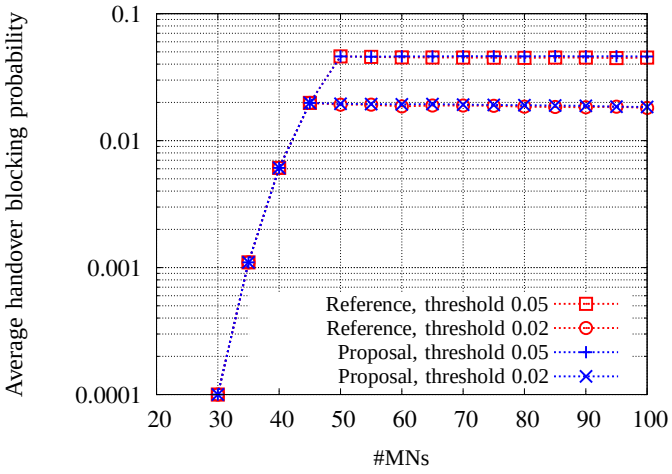


Figure 2. Handover blocking probability *versus* the number of MNs in Scenario 1.

Figure 3 presents results for the Scenario 2. It present similarities with Figure 2 but the curves gets stable sooner: from 30 MNs for the threshold *0.02* and from 35 MSs for the threshold *0.05*. This anticipation is due to the shorter total available bandwidth in the scenario studied. Thus, handover blocking probability increases faster, but it also gets stable in accordance with the established threshold.

Figure 4 shows results for latency in Scenario 1. With respect to the scheme in [11], we can observe that latency starts growing from 45 MNs for threshold *0.02*. Values in that curve are greater than those for threshold *0.05*, which starts growing from 50 MNs. As we have stated before, the stabilization of the blocking probability curve observed in Figure 2 implies the increase of the average latency. Also, there is a greater number of handover attempts when we use a lower threshold. Thus, the threshold *0.02* is more conservative and tends to make MNs wait for more time than those using threshold *0.05*. The lower the threshold is, the greater is the average latency. We can also observe in Figure 4 the impact of the proposed

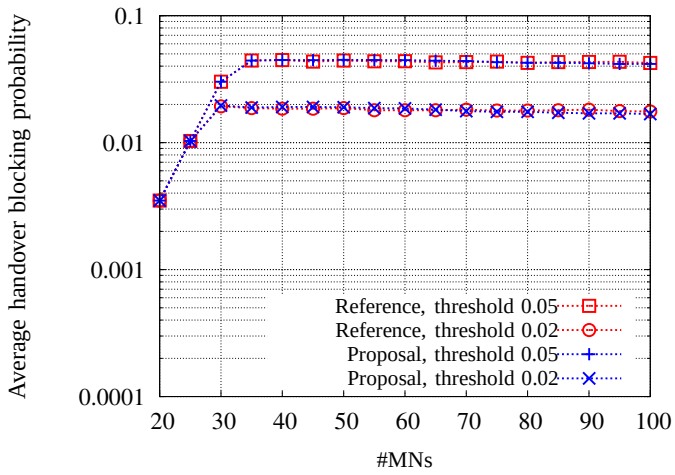


Figure 3. Handover blocking probability *versus* the number of MNs in Scenario 2.

solution on the latency curve. The curve becomes smoother than the curve that does not adopt the solution. For threshold *0.05*, the latency is 20% smaller in the case of 65 MNs and 28% smaller for 100 MNs. For threshold *0.02*, latency is 24% smaller for 65 MNs and 33% smaller for 100 MNs. The latency reduction is due to the proposed solution, which makes the delay between attempts more flexible. The exponential backoff also brought randomization to the scheme allowing MNs to try handover again sooner and in different periods of time, eventually reducing the total number of attempts.

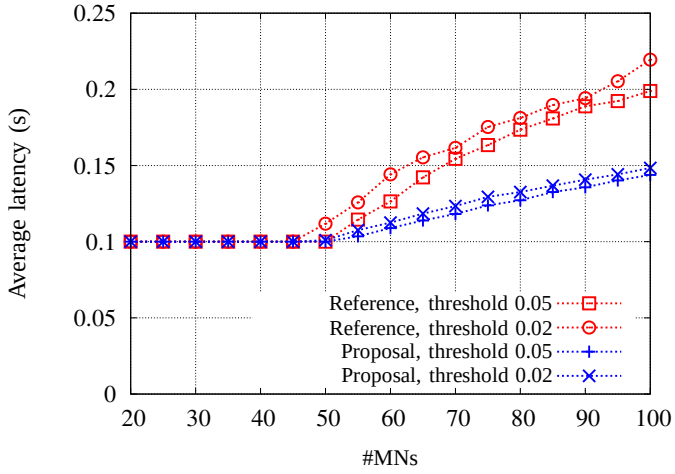


Figure 4. Latency *versus* the number of MNs in Scenario 1.

Figure 5 shows results for latency in Scenario 2. As in Scenario 1, the curve for threshold *0.02* has greater latency values than the one with threshold *0.05*. In [11], latency starts growing from 25 MNs for threshold *0.02* and from 30 MNs for threshold *0.05*. In Scenario 2, we also notice that there is a greater slope in latency as the number of MNs increases as shown in [11]. Greater latency values are expected because the total available bandwidth is shorter than in Scenario 1. However, the latency value is two times greater when the

 ISBN: 978-1-61208-318-6

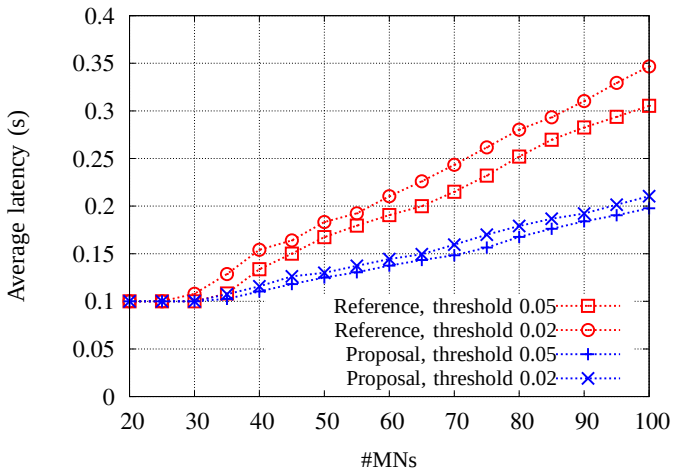


Figure 5. Latency *versus* the number of MNs in Scenario 2.

number of MNs reaches 60 for the threshold *0.02*. Regarding the same curve, we have 350 ms for 100 MNs. It is important to notice that more than two-thirds of this time is spent only in the handover decision in [11]. Figure 5 also shows that once again the proposed solution had the effect of reducing latency and easing the slope of the latency curve. For the threshold *0.05*, latency has a reduction of 29% for 65 MNs and 36% for 100 MNs. For the threshold *0.02*, we observe a reduction of 24% for 65 MNs and 40% for 100 MSs.

VII. Conclusion and Future Work

In this paper, we have proposed a policy for GVHO attempts. Our solution uses exponential backoff in order to allow a better distribution of handover attempts over time. Performance evaluations have shown that our proposal makes it possible to reduce handover latency and ease the slope of the latency curve as the number of MNs grows. In particular, results have shown that latency was reduced up to 40% in accordance with the scenarios evaluated. In future work, we will evaluate our proposal in other scenarios. We will take into account a varying number of PoAs, traffic data, different parameter values, and additional evaluation metrics. Also, we intend to include MIH queries in the solution design and to include the information gathering phase in performance evaluation. We are also planning to study the impact of our solution on other GVHO schemes.

References

[1] S. Lee, K. Sriram, K. Kim, Y. Kim, and N. Golmie, “Vertical Handoff Decision Algorithms for Providing Optimized Performance in Heterogeneous Wireless Networks,” IEEE Transactions on Vehicular Technology, vol. 58, no. 2, 2009, pp. 865–881.

[2] S. Zeadally and F. Siddiqui, “An Empirical Analysis of Handoff Performance for SIP, Mobile IP, and SCTP Protocols,” Wireless Personal Communications, vol. 43, no. 2, 2007, pp. 589–603.

[3] W. Shen and Q. Zeng, “Cost-function-based network selection strategy in integrated wireless and mobile networks,” IEEE Transactions on Vehicular Technology, vol. 57, no. 6, 2008, pp. 3778–3788.

[4] E. Stevens-Navarro, Y. Lin, and V. Wong, “An MDP-Based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks,” IEEE Transactions on Vehicular Technology, vol. 57, no. 2, 2008, pp. 1243–1254.

[5] K. Taniuchi, Y. Ohba, V. Fajardo, S. Das, M. Tauil, Y. Cheng, A. Dutta, D. Baker, M. Yajnik, and D. Famolari, “IEEE 802.21: Media Independent Handover: Features, Applicability, and Realization,” IEEE Communications Magazine, vol. 47, no. 1, 2009, pp. 112–120.

[6] C. Cicconetti, F. Galeassi, and R. Mambrini, “Network-assisted Handover for Heterogeneous Wireless Networks,” in Proc. of the IEEE GLOBECOM Workshops, Miami, 2010, pp. 1–5.

[7] K. Andersson, A. Forte, and H. Schulzrinne, “Enhanced Mobility Support for Roaming Users: Extending the IEEE 802.21 Information Service,” in Proc. of the 8th International Conference on Wired/Wireless Internet Communications, Berlin, 2010, pp. 52–63.

[8] I. Kim and Y. Kim, “Performance Evaluation and Improvement of TCP Throughput over PFMIPv6 with MIH,” in Proc. of the 12th IFIP/IEEE International Symposium on Integrated Network Management, Dublin, 2011, pp. 997–1004.

[9] M. Zekri, B. Jouaber, and D. Zeghlache, “A Review on Mobility Management and Vertical Handover Solutions over Heterogeneous Wireless Networks,” Computer Communications, vol. 35, no. 17, 2012, pp. 2055–2068.

[10] H. Jeong, J. Choi, H. Kang, and H. Youn, “An Efficient Group-Based Channel Scanning Scheme for Handover with IEEE 802.16e,” in Proc. of the 26th IEEE International Conference on Advanced Information Networking and Applications Workshops, Fukuoka, 2012, pp. 639–644.

[11] W. Lee and D. Cho, “Enhanced Group Handover Scheme in Multi-Access Networks,” IEEE Transactions on Vehicular Technology, vol. 60, no. 5, 2011, pp. 2389–2395.

[12] X. Cai and F. Liu, “Network Selection for Group Handover in Multi-access Networks,” in Proc. of the IEEE International Conference on Communications, Beijing, 2008, pp. 2164–2168.

[13] D. Niyato and E. Hossain, “Dynamics of Network Selection in Heterogeneous Wireless Networks: an Evolutionary Game Approach,” IEEE Transactions on Vehicular Technology, vol. 58, no. 4, 2009, pp. 2008–2017.

[14] S. Lei, T. Hui, and H. Zheng, “Group Vertical Handover in Heterogeneous Radio Access Networks,” in Proc. of the 72nd IEEE Vehicular Technology Conference Fall, Ottawa, 2010, pp. 1–5.

[15] S. J. Bae, M. Y. Chung, and J. So, “Handover Triggering Mechanism Based on IEEE 802.21 in Heterogeneous Networks with LTE and WLAN,” in Proc. of the International Conference on Information Networking (ICOIN), Barcelona, 2011, pp. 399–403.

[16] M. Q. Khan and S. H. Andresen, “PoA Selection in 802.11 Networks Using Media Independent Information Server (MIIS),” in Proc. of the 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Fukuoka, Japan, 2012, pp. 454–459.

[17] Y.-H. Liang, B.-J. Chang, and C.-T. Chen, “Media Independent Handover-based Competitive On-Line CAC for Seamless Mobile Wireless Networks,” Journal Wireless Personal Networks, 2011.

[18] L. Shan, F. Liu, L. Wang, and Y. Ji, “Predictive Group Handover Scheme with Channel Borrowing for Mobile Relay Systems,” in Proc. of the International Wireless Communications and Mobile Computing Conference, Crete Island, 2008, pp. 153–158.

[19] B. Kwak, N. Song, and L. Miller, “Performance Analysis of Exponential Backoff,” IEEE/ACM Transactions on Networking, vol. 13, no. 2, 2005, pp. 343–355.

[20] Transport Systems Generic Requirements (TSGR): Common Requirements. GR-499, Issue 2, Telecordia Report, 2009.

[21] M. K. Karray, “Evaluation of the Blocking Probability and the Throughput in the Uplink of Wireless Cellular Networks,” in Proc. of the International Conference on Communications and Networking, Tozeur, Tunisia, 2010, pp. 1–8.

[22] C. Chekuri, K. Ramanan, P. Whiting, and L. Zhang, “Blocking Probability Estimates in a Partitioned Sector TDMA System,” in Proc. of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications, Boston, USA, 2000, pp. 28–34.

[23] G. Haring, R. Marie, R. Puigjaner, and K. Trivedi, “Loss Formulas and Their Application to Optimization for Cellular Networks,” IEEE Transactions on Vehicular Technology, vol. 43, no. 3, 2001, pp. 664–673.

 ISBN: 978-1-61208-318-6

Fault Tolerance in Area Coverage Algorithms for Limited Mobility Sensor Networks

Mark Snyder, Sriram Chellappan
Dept. of Computer Science
Missouri University of Science & Technology
Rolla, Missouri, USA
Email: marksn_ms@msn.com, chellaps@mst.edu

***Abstract*—In sparse deployments of mobile sensors, the mobility of sensors is required to search the coverage area in an attempt to achieve polling—complete, possibly repeated, area coverage over time. Mobile sensor platforms are vulnerable to a variety of hazards during normal operation. When sensors lose the ability to mobilize due to mechanical failure, environmental factors, or simply exhausting their energy source, coverage effectiveness can be seriously impacted. Ideally, algorithms should adjust their behavior to compensate for failure modes in order to avoid areas statically covered by disabled sensors as well as adjusting their behavior to cover the areas assigned to sensors that are no longer able to mobilize. In this work, we demonstrate the effects of disabled mobility on area coverage algorithms due to their inability to adjust behavior, and suggest mitigation strategies and the impact on improving coverage in the face of disabled mobility.**

***Keywords*—Mobile Sensors; Sensor Networks; Disabled Mobility; Algorithms**

I. INTRODUCTION

There are numerous factors driving increased attention to usage of automated sensor drones that consist of a hardware platform with onboard command/control, sensors, and effectors that provide for mobility. Human safety factors have long been a primary motivator for interest in employing robots for a variety of tasks where humans would prefer not to be. Also, mobile sensor hardware platforms have seen many recently advances, such as higher computation power, relatively lower weight, and lower power requirements that allow drones to carry much more computational capacity and payload or stay deployed and functional for longer periods of time. More and more, these platforms are more readily available, as the benefits of mass production of commercially designed systems are realized.

Mobile sensors are physical devices that are subject to observable failure rates. One of the most common problems for mobile sensors is that the effectors that provide for the mobility function of the platform fail [1], or that terrain or other issues cause the mobile sensor to become stuck while all other features of the platform continue to function as normal.

It is intuitive that one requirement of an algorithm being analyzed is that for the algorithm to function, sensors must be aware to some degree of the relative position and mobility restrictions (failure modes) impacting other sensors, otherwise there is no way they can (nor any reason for them to) alter their behavior. However, mobile sensors need not support full localization. Awareness of the failure modes of other sensors allows the network to be fault-tolerant, self-healing, and to dynamically change from initially homogeneous to heterogeneous with respect to mobility as agents adopt different roles (whether by choice or as observed).

Further, we acknowledge that various subsystems of a wireless sensor exhibit different energy requirements than others. Since our primary priority is to maintain a given network quality of service, we are required to utilize the mobility feature of the sensors. What we will establish is a schedule, whereby a certain number of sensors lose their mobility but retain their other functions, and will analyze the extent to which other sensors are able to compensate for this failure mode by altering their movement strategy to preserve the required quality of service while avoiding the area covered by the disabled sensors.

Given an objective measurement for lifetime and effectiveness of a sensor network, we explore the effects of disabled mobility on these metrics. From this work, we can see that as sensors lose their mobility, they become, in essence, static rather than mobile sensors. When the deployment and/or the algorithm that governs their behavior ensures that mobile sensors are spread sufficiently to adequately cover the search area, then coverage impact can be minimal. Conversely, when the deployment is concentrated or the algorithm fails to spread sensors prior to many of them failing, coverage effectiveness is vulnerable. Additionally, when mobility fails but other functions, such as communication, continue to operate, a sensor can communicate misleading intentions to other sensors. In essence, this can have outcomes similar to an attack where blind spots are created. The disabled sensor cannot navigate to the intended location to cover it, and other mobile sensors choose not to go there because they believe it is already covered.

In this paper, we examine related work in the areas of limited mobility, reliability analysis, and fault tolerance. After a brief survey of mobile sensor platforms, their mobility limitations, and their vulnerabilities to device failure, we describe the problem of assessing the impact of disabled mobility on coverage algorithms. We define the reliability model we will use, describe the simulation platform and parameters selected for our analysis, several coverage algorithms that are used for comparison, and then present findings, conclusions, and future work.

II. RELATED WORK

Fault tolerance in mobile sensor networks, the topic of algorithms that behave in a way that tolerates failure modes while still cooperatively pursuing a goal, can be found [2], [3], [4], but the specific topic of fault tolerance with respect to limited mobility sensor networks and how failures affect coverage performance has not.

One area where limited mobility affects performance of a mobile sensor network is when sensors are deployed for blanket coverage. The lifetime of such networks, and algorithms for preserving/extending the lifetime, has been extensively studied.

 ISBN: 978-1-61208-318-6

An example uses redundant sensors in a dense deployment so that as sensors fail other sensors in the same region can wake up and take over for the missing sensor [5]. There are formulae that can be used to objectively assess the expected lifetime of a network of sensors [6]. Also, many works have explored the idea of optimized message routing for sensor networks in the event that some sensors fail, so that messages can be routed through other paths [7]. However, there has been little work devoted to the area of exploring what happens when a sensor continues to function even though it is no longer able to navigate. This is a real concern given that in many real world sensor platforms, energy requirements for mobility account for a large portion relative to that required for sensing and communication. The closest analogy is the study of hybrid sensor networks, where the term hybrid refers to the fact that sensors are non-homogeneous: some are static, and others are mobile [8], [9]. However, no works were found that examine the problem of sensors failing according to a predetermined failure model, additionally that the failure is limited to the mobility feature of the device, and relates this to the impact on coverage effectiveness.

As further background, a number of works examine sensor network lifetime from a rather fatalistic point of view, expressing desire to describe and understand an inevitable upper bound on the utility of a network of sensors [10].

It was shown that effectors–devices that perform actuation (including mobility), such as the motor, appendages, treads/wheels, and related connections—was observed to account for 35% of mobile robot failure, the largest single reason for failures [1]. This makes consideration of the problem of maintaining coverage quality of service (among other goals) using cooperating sensors in the network, an important aspect of mobile sensor network research.

Various works explore fixed deployments (no mobility following the initial deployment of the sensors), where mobility is not a concern toward energy constraints on the lifetime of the network. In a dense sensor deployment scenario, more sensors are deployed than are required to cover an area, and when a sensor fails, other sensors use a protocol to decide which sensor wakes up and takes the place of sensing the missing area. In some cases, sensors have a limited ability to exercise mobility, and can move closer to the hole in coverage in order to adjust for the missing sensor.

III. Reliability in Sensor Platforms

Numerous mobile sensor platforms have been in development in recent decades, including ground-based, lift-based, buoyancy-based, and space-based. Ground-based platforms (sometimes referred to as UGV's [11]) can be tiny weighing only a few centimeters/grams, using battery-powered micro circuitry, or as large as automobiles weighing tons using internal-combustion engines. These platforms are subject to a variety of mechanical failures, are vulnerable to obstacles found on the ground in the environments in which they operate, as well as terrain variations and pitfalls. The energy source (weight and conservation) is a major factor limiting the mobility of these platforms.

Lift-based, or aerial, platforms are devices that employ the physics of lift in order to remain in a state where controlled mobility is possible. The size of these devices can range from small, hand-held devices, up to large military/commercial aircraft. Identifying aircraft with a low Reynolds number provides a way to construct devices that are useful for lab research [12]. The Reynolds number can be expressed as shown in (1), where ρ is the air density, L is airfoil length, υ is velocity, and μ is the viscosity of the substance through which the device moves. Utilizing this formula allows researchers to create small, lightweight devices that can move slowly and stay aloft for longer periods of time. However, a challenge faced by lift-based platforms is that they must expend energy to maintain continual lift. Mechanical failures are often fatal due to engineering the devices to use minimal structural material to minimize fuel requirements and allow for more payload, which in turn makes the devices more fragile than their ground-based cousins.

TABLE I: Approximate weight and buoyancy of various substances

Substance	Weight	Buoyancy
Air	$1.2256 Kg/m^3$	—
Hydrogen (H)	$0.0857 Kg/m^3$	$1.1399 Kg/m^3$ (H v. Air)
Helium (He)	$0.1691 Kg/m^3$	$1.0565 Kg/m^3$ (He v. Air)
Water (H_2O)	$988.2 Kg/m^3$ @ $20\,^\circ$C	$0.24875 Kg/m^3$ (Air v. Water)

$$Re = \frac{\rho L \upsilon}{\mu} = \frac{\rho \upsilon^2}{\frac{\mu \upsilon}{L}} = \frac{inertia}{viscosity} \quad (1)$$

Buoyancy-based platforms solve many of the problems faced by ground-based and lift-based platforms. Examples of this type of platform include blimps and boats. These devices can be tiny to enormous commercial tanker ships. They are characterized by the ability to maintain a stable navigational state for long periods of time (indefinitely, barring other issues, such as leaks), and the ability to support a much larger payload over time than lift-based or even ground-based platforms. Vulnerabilities include currents in the substance (typically water or air) in which the devices operate, weather, and obstacles. When we consider the relative densities of substances, we can approximate the weight and buoyancy for devices utilizing these substances as shown in Table I. Thus, the desired payload can be defined and the device characteristics tailored to fit.

Space-based platforms have been in use for nearly six decades. These devices must use some combination of lift, buoyancy, and thrust in order to place the device in "space" where it is capable of remaining aloft in a state where its navigational attributes are governed by inertia and orbital mechanics, and where the viscosity becomes negligible. Such platforms are vulnerable to impacts with other objects traveling at very high velocities, orbital decay causing atmospheric reentry, cosmic rays and radiation, extreme heat and cold, in addition to standard mechanical failures with rare opportunities for service/repair.

Reliability analysis studies the probability of devices performing the function for which they were designed over a period of time within specified parameters. In [13], we see analysis of failure rate models for devices. We describe time-to-failure as a probability density function (PDF) or cumulative density function (CDF). The probability of failure over time $F(t)$ may be expressed mathematically as shown in (2). Alternatively, the probability of reliability over time $R(t) = 1 - F(t)$. The function $f(x)$ represents a distribution of failures over time, and the interval between t_0 and t_1 is the period of time during which the devices are observed.

 ISBN: 978-1-61208-318-6

$$F(t) = Pr\{T : t_0 \leq x < t_1\} = \int_{t_0}^{t_1} f(x)dx \qquad (2)$$

Failure rates for these various platforms are becoming more widely available as technologists spend more time settling on one design and tracking its reliability [14], [4], [1], [15], [11], [16]. As these platforms become more common, we will be able to develop more applicable and accurate failure rate models for each type of platform.

IV. Problem Details

We define the coverage field as a region that is observed as a plane to sensors. A set of mobile sensors is deployed using a deployment function. In this paper, we focus on two deployment schemes. First, a purely stochastic means of evenly distributing sensors throughout the field. Second, a stochastic method that produces a Gaussian approximation of a Poisson distribution around a point, as if the sensors might have been dropped from an aircraft and dispersed organically at various distances and orientations relative to the drop point.

We focus on sparse deployments in which the number of sensors n is defined in (3), A is the area of the coverage field and r is the sensing range. This relationship ensures that the number of sensors being lower than required to make blanket coverage possible. This allows us to focus on finding solutions to the problem of polling–minimizing detection time for any events in the coverage field, while maximizing the number of times we can poll all points in the coverage field over a given period of time.

At this point we define *polling frequency* as the number of times the entire coverage field is sensed in a given time period.

$$n < \frac{4A}{3\sqrt{3}r^2} \qquad (3)$$

In balanced deployments, the problems shift from finding solutions that minimize the time to achieve (and maintain) blanket coverage, whereas in dense deployments the problems shift from the challenge of providing polling to one of maintaining blanket coverage or redundant blanket coverage. When a sensor becomes mobility-disabled in balanced to dense deployments, the network's ability to maintain blanket coverage for a length of time can be shortened as sensors ultimately fail completely and the inability of other sensors to take their place causes coverage holes.

The mobility of the sensors is considered to be limited, in that there is a probability that at a certain time interval from the drop time that a given sensor might suddenly lose the ability to move. This simulates the lifetime of the mobility feature of the sensor. However, the sensor continues to be able to take measurements of its environment from this location. The number of sensors that have failed over time is controlled such that it follows a probability density function. As an example, the number of sensors that have failed over time might look like one of the models shown in Figure 1.

The "bathtub curve" model for failure rates has been described [13]. In this model, numerous initial failures are observed, followed by a stable period where few failures occur, and finally a period of time where devices succumb to the useful lifetime of any of a number of their components causes a relatively higher failure rate to account for a majority of the remaining devices. While this model describes the failure

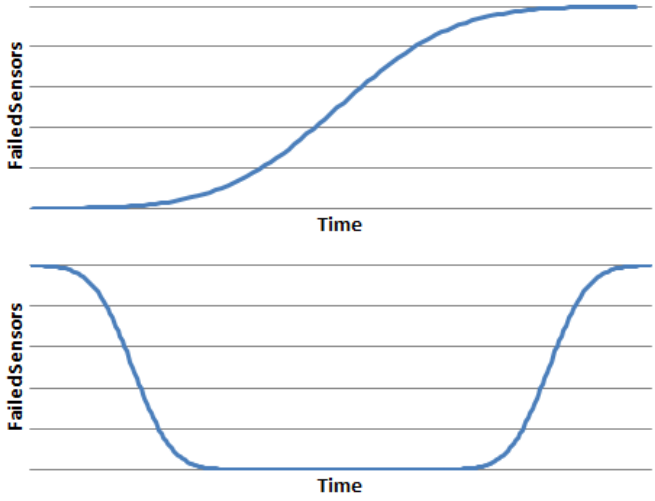


Figure 1: Cumulative distribution function (top) illustrating how we will distribute the disabling of mobility of sensors over time, plus Bathtub Curve (bottom) showing another well known failure rate model

rate of an entire population of devices over time, this model may not be as useful for studying the effects of failures of a specific set of devices in use in the field. This is due to the fact that initial testing and quality control measures will identify defective products prior to deployment, and devices may be replaced in the field long before they actually fail during use. For this reason, we focus on the cumulative distribution function (CDF) model for our simulations that assumes no failures initially, but a growing number of failures as the simulation progresses, followed by a few sensors that fail later.

V. Algorithms

In our simulations, we chose several algorithms to analyze and compare with varying sensor count (sparsity), two deployment schemes (random and Gaussian around a point), with and without applying the schedule of disabled mobility, and in some cases with and without modifications to the algorithm to mitigate the effects of disabled mobility.

The Random Walk and Random Direction Walk [17] are used. The Random Walk algorithm has sensors simply choose at each opportunity in time a random direction to move, whereas the Random Direction Walk algorithm chooses a random direction up front and continues that direction for the life of the simulation. Although these algorithms include no cooperative features, nor do they attempt to avoid gaps or redundant coverage in any way, they provide a good baseline for comparison to other algorithms.

The Proxy [18] and WGB [19] algorithms were also used. The distributed Proxy-based algorithm involves both static and mobile sensors which bid for new locations in order to heal holes in coverage. The WGB algorithm, also a distributed heuristic-based approach, uses an internal tile-coloring model to merge data about what areas nearby are not covered (white), have sensors nearby that could cover (gray), and areas that are already occupied (black), in order to identify the location of highest need. We focus on WGB in order to eliminate the variable of static sensors from disabled mobility sensors. Other

 ISBN: 978-1-61208-318-6

algorithms, such as Virtual Force (VF) [20], [21] were also examined.

VI. Effects of Disabled Mobility

We anticipate a few challenges that will affect coverage efficiency in the face of disabled mobility. First is the fact that a disabled sensor is sitting in one place sensing the area around it and other sensors that pass through this area will duplicate coverage resulting in a loss of efficiency. Second, the disabled sensor could have been expected to have covered a portion of the field itself had it not become disabled, thus, other sensors may need to adjust their movement plan in order to cover the area excluded by the loss of the disabled sensor. Another factor of interest is the probability of detection and the detection time delay behavior in the presence of the disabled mobility schedule.

With some algorithms, disabled sensors may in fact mislead other sensors about their intended mobility plan and affect coverage in ways that would not be seen if the sensor were to completely fail.

Let us consider a scenario where we assume a random distribution of sensors across the search area, where sensors have unlimited mobility (range). The sensing distance is configured so that this is a sparse to balanced deployment (i.e., the ratio of sensor range to number of sensors relative to search area precludes blanket coverage). The goal is to maximize area coverage over time (or synonymously to minimize detection delay). We examine two reference algorithms. First, the Random Walk algorithm, where each mobile sensor starts exploring the area in random moves. Second, we examine the Random Direction Walk algorithm, where each sensor begins by picking an initial direction and continually moves straight in that direction indefinitely.

Analyzing the results of a schedule of sensor mobility disability surfaced a challenge with this scenario. Examining the initial deployment, we observe a random distribution of mobile sensors throughout the search area. Also, at any time in the future, a snapshot of the region also shows a distribution with no less random features than the initial deployment. Despite the fact that the point at which a given sensor becomes disabled is according to a pre-determined schedule, the location at which it resides when it becomes disabled is again no less randomly distributed than the initial deployment. Thus, observing simulation of this scenario over time as seen in Figure 2 shows that although the performance isn't great at any point in time throughout the runs, disabling the mobility of the sensors doesn't hurt the algorithm in an interesting or unexpected way.

Using the Random Direction Walk algorithm produces analogous results, and both algorithms are consistent even when the number of sensors is varied. Figure 3 shows a consistent drop-off in polling frequency with random direction walk across a variety of sensor counts. Polling frequency eventually flattens due to the sparse deployment density and the fact that disabled sensors fail to iteratively cover the area over time. This produces an equivalent increase in average detection delay as more and more of the area must be polled by a decreasing population of sensors with mobility.

When we examine the effects of disabled mobility on the WGB algorithm, we see a consistent drop in coverage performance, and falling to as much as 20% loss of coverage as sensors begin to lose mobility. Figure 4 illustrates how many

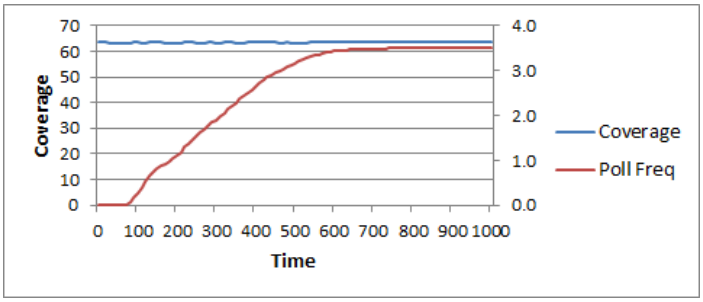


Figure 2: Random walk coverage and polling frequency over time

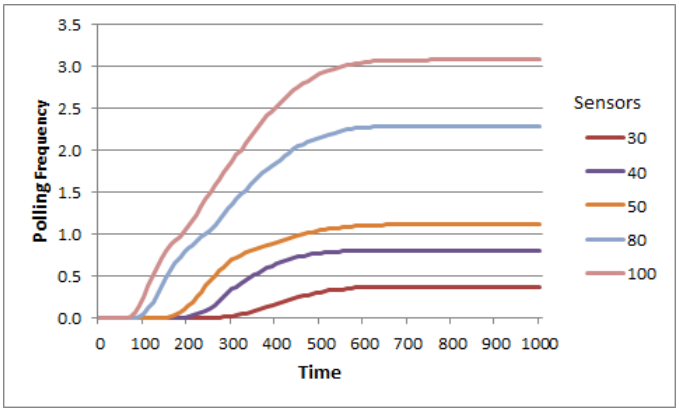


Figure 3: Random direction walk polling frequency over time

percentage points coverage drops with the WGB algorithm in particular, simply by adding the schedule of disabled mobility over time. This is actually quite good considering that the algorithm pushes sensors from their initial deployment to a balanced coverage of the area rather quickly.

We can also see from Figure 5 a representation of the performance of the WGB algorithm in terms of the coverage percentage over time for a varying number of sensors. In order to properly interpret the sparsity of the deployment given the specified sensor counts, we refer once again to (3) with a configuration of coverage area and sensor range that results

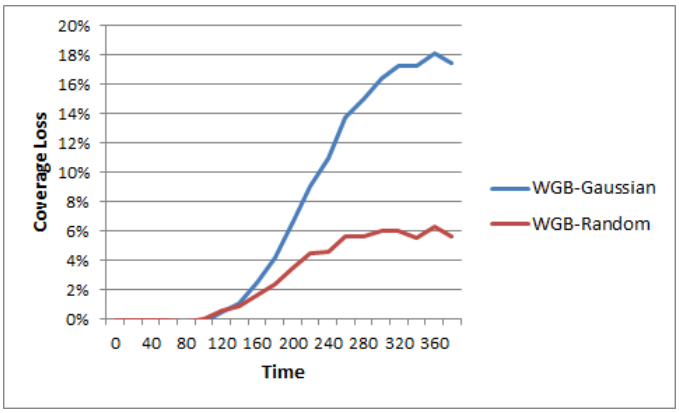


Figure 4: The number of percentage points lost by adding disabled mobility to WGB algorithm

TABLE II: Max coverage % for various sensor counts

Sensors	Max %
25	39.26
30	47.12
35	54.97
40	62.83
50	78.53
80	125.66
100	157.07

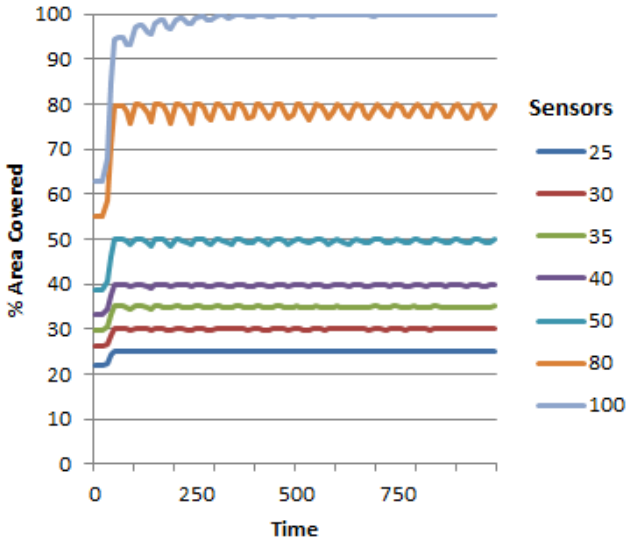


Figure 5: WGB algorithm, coverage area over time varying sensor count

in a balanced deployment would require a value of $n \approx 76.98$. Given the size of the coverage field in these simulations relative to sensor range and number of sensors, we can see the computed maximum coverage achievable for various sensor counts. These are upper bounds, and rely on none of the sensors overlapping areas covered by other sensors. In practice, for our distributed algorithms to consistently achieve polling, the number of sensors remains at 80 or below.

Regarding the problem of disabled mobility causing potentially misleading information being broadcast to other sensors, we see that there is a clear impact to coverage efficiency. By making a small adjustment to the WGB algorithm to detect this failure mode, plus a behavior change when the failure mode occurs such that sensors refrain from broadcasting an intention to move that will not occur, we see an improvement in both Gaussian and Random deployment modes of a full percentage point. Figure 7 shows the improvement for one configuration. As shown, the improvement begins as sensors start to fail. When a growing number of sensors are unable to move, the ability for the WGB algorithm to continue to cooperatively explore the coverage field without gaps or significant redundant coverage becomes apparent as compared to Random Walk, Random Direction Walk, and other algorithms.

VII. Conclusions

One thing we can observe from these results is that coverage algorithms that do a good job of quickly reaching a

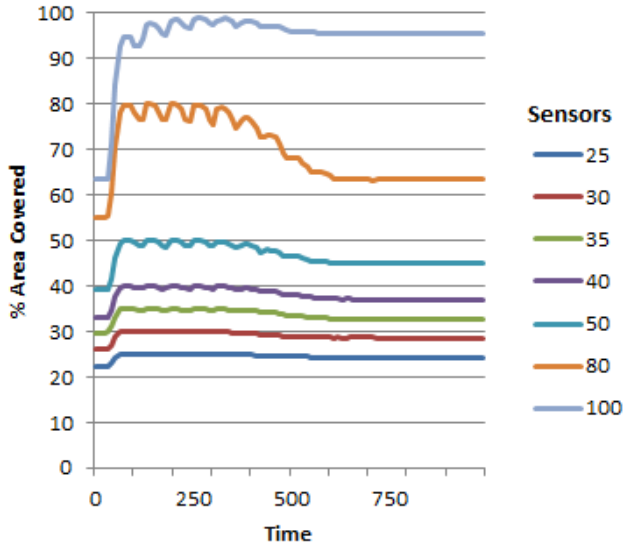


Figure 6: WGB algorithm with disabled mobility, coverage area over time varying sensor count

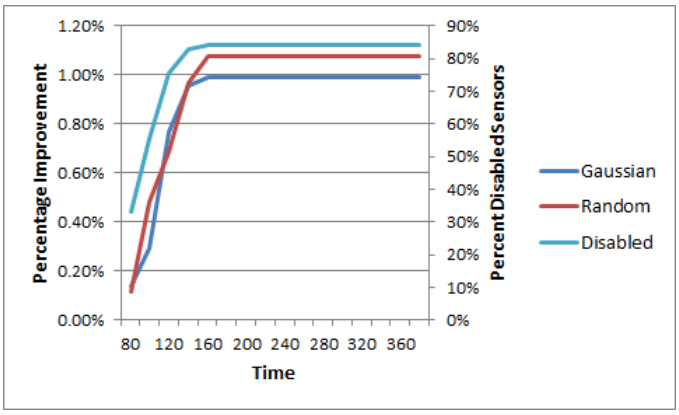


Figure 7: Percentage improvement in coverage efficiency by detecting failure mode

desirable location from their initial deployment, cooperate to avoid gaps and redundant coverage, and continue to leverage what mobility is available throughout the sensor network, produce better results as sensors lose their mobility than algorithms that rely on statically touring or other more methodical means of exploring and covering their environment. The WGB algorithm, for example, saw only minimal degradation of coverage quality of service, and performed well in sparse to balanced deployments in the face of a schedule for disabled mobility.

With extremely sparse deployments, sensors that are mobile come into contact with disabled sensors less often. In these scenarios, we observe through simulations that algorithms, such as random direction walk have less of an impact than algorithms that tour an established territory, because in the latter case, once a sensor becomes disabled, there is no sensor to cover that sensor's territory. As the deployment becomes less sparse, algorithms that try to avoid one another are more

 ISBN: 978-1-61208-318-6

vulnerable to being misled by disabled sensors that continue to broadcast their intentions to move, but never do.

The disabled mobility problem has particular significance because, as defined, the outcome can be demonstrated clearly as an extension of prior proven simulation techniques. The proposed approach introduces a factor whereby sensors become immobile at various rates over time. When the coverage algorithm is to pick a random direction, then sensors will disregard the location and coverage provided by disabled sensors and will proceed to duplicate coverage. When algorithms avoid those areas, coverage effectiveness can be shown to increase. As more sensors become disabled, coverage becomes degraded, as we have shown.

VIII. Future Work

Communication protocols have been extensively studied from a number of perspectives. However, there is potential for augmenting these protocols to transmit failure modes along with existing packets in order to allow distributed algorithms to reactively modify their behavior to make the sensor network self-healing fault tolerant. For example, consider a mobile sensor that is able to transmit a set of p failure modes $F = \{f_0, f_1, ..., f_p\}$, where $f_i \in \{0, 1\}$. Each failure mode represents a test result from an onboard sensor that tests an aspects of the sensor's normal operational state and report about what portions of the sensor are working (1) or not (0). If we assume a homogeneous set of mobile sensors, then each sensor would understand what aspect of its counterpart was malfunctioning by reading this stream contained within a packet sent according to the communication protocol used by the mobile sensors. Thus, we could develop algorithms that adjust their navigation choices after filtering data from other sensors. Such algorithms would not be as susceptible to being misled by the communicated intended actions of other sensors.

Acknowledgments

This work was supported in part by National Science Foundation (NSF) under Grant No. 1254117 and 1205695. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. Carlson and R. R. Murphy, "Reliability analysis of mobile robots," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1, pp. 274–281, IEEE, 2003.

[2] G. Hoblos, M. Staroswiecki, and A. Aitouche, "Optimal design of fault tolerant sensor networks," in *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, pp. 467–472, IEEE, 2000.

[3] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *Network, IEEE*, vol. 18, no. 4, pp. 36–44, 2004.

[4] M. L. Visinsky, J. R. Cavallaro, and I. D. Walker, "Robotic fault detection and fault tolerance: A survey," *Reliability Engineering & System Safety*, vol. 46, no. 2, pp. 139–158, 1994.

[5] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.

[6] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *Communications Letters, IEEE*, vol. 9, no. 11, pp. 976–978, 2005.

[7] Y. Chen, Q. Zhao, V. Krishnamurthy, and D. Djonin, "Transmission scheduling for optimizing sensor network lifetime: A stochastic shortest path approach," *Signal Processing, IEEE Transactions on*, vol. 55, no. 5, pp. 2294–2309, 2007.

[8] W. Wang, V. Srinivasan, and K. C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *MOBICOM*, pp. 39–50, 2007.

[9] D. Wang, J. Liu, and Q. Zhang, "Probabilistic field coverage using a hybrid network of static and mobile sensors," in *Quality of Service, 2007 Fifteenth IEEE International Workshop on*, pp. 56–64, IEEE, 2007.

[10] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 5, 2009.

[11] J. Carlson and R. R. Murphy, "How ugvs physically fail in the field," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 423–437, 2005.

[12] J.-D. Nicoud and J.-C. Zufferey, "Toward indoor flying robots," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, pp. 787–792, IEEE, 2002.

[13] C. Lai, M. Xie, and D. Murthy, "Ch. 3. bathtub-shaped failure rate life distributions," in *Advances in Reliability* (N. Balakrishnan and C. Rao, eds.), vol. 20 of *Handbook of Statistics*, pp. 69 – 104, Elsevier, 2001.

[14] J. R. Cavallaro and I. D. Walker, "A survey of nasa and military standards on fault tolerance and reliability applied to robotics," in *NASA CONFERENCE PUBLICATION*, pp. 282–282, NASA, 1994.

[15] J. Carlson, R. Murphy, and A. Nelson, "Follow-up analysis of mobile robot failures," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 5, pp. 4987–4994 Vol.5, IEEE, 2004.

[16] S. Stancliff, J. M. Dolan, and A. Trebi-Ollennu, "Towards a predictive model of robot reliability," 2005.

[17] E. Atsan and Ö. Özkasap, "A classification and performance comparison of mobility models for ad hoc networks," in *Ad-Hoc, Mobile, and Wireless Networks*, pp. 444–457, Springer, 2006.

[18] G. Wang, G. Cao, and T. La Porta, "Proxy-based sensor deployment for mobile sensor networks," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 493–502, IEEE, 2004.

[19] M. Snyder, S. Chellappan, and M. Thakur, "Exploratory coverage in limited mobility sensor networks," in *Network-Based Information Systems, 2013. NBIS'13. International Conference on*, IEEE, September 2013.

[20] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*, pp. 299–308, Springer, 2002.

[21] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, pp. 1293–1303, IEEE, 2003.

 ISBN: 978-1-61208-318-6

Performance Comparison of IPv6 Multihoming and Mobility Protocols

Charles Mugga, Dong Sun, Dragos Ilie
School of Computing
Blekinge Institute of Technology (BTH)
Karlskrona, Sweden
E-mail: {chmu11, sudo11}@student.bth.se, dragos.ilie@bth.se

***Abstract*—Multihoming and mobility protocols enable computing devices to stay always best connected (ABC) to the Internet. The focus of our study is on handover latency and rehoming time required by such protocols. We used simulations in OMNeT++ to study the performance of the following protocols that support multihoming, mobility or a combination thereof: Mobile IPv6 (MIPv6), Multiple Care-of Address Registration (MCoA), Stream Control Transmission Protocol (SCTP), and Host Identity Protocol (HIP). Our results indicate that HIP shows best performance in all scenarios considered.**

Keywords-IPv6; mobility; multihoming; performance.

I. INTRODUCTION

Modern computing devices such as laptops, tablets, smart phones, PCs and broadband routers are typically equipped with multiple networks interfaces (e. g. , 3G, WiFi) that enable users to stay always best connected (ABC) to the Internet [1]. What *best connected* means is intimately tied to the needs of a particular user: for some, it means being connected over the interface that offers the highest data rate, while for others connectivity during movement may be more important. Such scenarios are supported by cooperation between multihoming and mobility management protocols at layer 3 (L3) and layer 2 (L2) handover mechanisms [2].

More specifically, horizontal handover allows a mobile node (MN) to change its link-layer point of attachment among networks using the same radio access technology (RAT). Similarly, vertical handover enables nodes to switch between networks based on different RATs. We will refer to these handovers as *L2 handovers* when they are transparent to L3 and above.

In certain situations, a node can be forced to reconfigure its IP address after a handover. For example, this happens when the handover occurs between networks under different administrative domains, where each domain manages its own set of network prefixes. In this situation, the handover requires assistance from L3 (i. e. , it is not longer transparent to the network layer). This type of scenario is called a *L3 handover* and is handled by mobility management protocols.

Multihoming is the ability to be simultaneously connected to multiple (home) networks. In practice, this means that each network interface is assigned an IP address from a different network, or that one interface is assigned multiple IP addresses corresponding to different networks. Benefits of multihoming include fault tolerance, load sharing, and bandwidth aggregation. The fault tolerance scenario, where communication over an unreachable network is reconfigured to go over another home network, is similar to a L3 handover. However, here we will use the term *rehoming* to emphasize that multihoming is handling this scenario.

Mobility and multihoming are generally considered as two separate concepts and thus are handled by different protocols. However, they both propose a mechanism for session survivability, which can be used to provide seamless connectivity. In our study, we focus on the performance of host-based mobility and multihoming protocols with emphasis on time to recover from link failures. The type of failures addressed here are due to node mobility or caused by stopped or failing router interfaces. Host-based mobility means that the MN is fully involved in mobility-related signaling. This is in contrast to network-based mobility, where dedicated entities are in charge of signaling and no mobility-specific features are required for MNs.

This paper is organized as follows. Section II provides an overview of several multihoming and mobility protocols (i. e. , HIP, MIPv6, SCTP, and MCoA), in terms of their modes of operation, benefits and drawbacks. Related work is described in Section III. Section IV discusses the basics of our simulation testbed and the particular simulation scenarios used here. Section V defines the performance metrics relevant to our study (i. e. , handover latency and rehoming time). Section VI elaborates on the simulation results. Finally, Section VII provides a summary and proposes future work to improve the performance of the studied multihoming and mobility protocols.

II. MOBILITY AND MULTIHOMING PROTOCOLS

This section provides an overview of IPv6 mobility and multihoming management protocols that were part of our study. Our main selection criteria was the availability of the protocol as OMNeT++ simulation model. A more comprehensive list of mobility and multihoming protocols can be found in [3]. Based on the functionality supported, each studied protocol was allocated to one of the following three categories: mobility management, multihoming management and combined multihoming-mobility management.

A. Mobility Management

The Mobile IPv6 (MIPv6) protocol was designed and incorporated in IPv6 during the base specification of IPv6, thus providing L3 integrated mobility management [4]. MIPv6 introduces a new element in the network architecture, the home agent (HA), which is responsible for maintaing communication

 ISBN: 978-1-61208-318-6

while the MN is visiting a foreign network. MIPv6 enables mobility by requiring a MN to use two addresses: the home address (HoA) and the care-of address (CoA) [5]. The HoA is a unique address from the home network address space. Its purpose is to be used as node identifier for the MN. When the MN visits a foreign network, it is assigned a CoA from the address space used by that network. The MN informs its HA whenever the CoA changes. The CoA identifies the topological location of the MN in the network graph, allowing packets to be routed to it.

Typically, a correspondent node (CN) always uses the HoA as destination address when sending data to the MN. The packets are received by the HA, which forwards them to the MN's CoA. The HA forwards also packets going in the opposite direction — from the MN to the CN. However, when both the MN and CN support route optimization, they can communicate directly using the CoA [6][7].

The strongest asset of MIPv6 is that it rests on over two decades of research and experimentation. As a result the protocol is mature and available for many platforms. Simultaneous L3 handover of the MN and CN is also supported. Some of MIPv6's drawbacks are the reliance on a third entity, the HA, high signaling overhead and some security issues related to return routability [5][8].

B. Multihoming Management

The Stream Control Transmission Protocol (SCTP) is a reliable connection-oriented layer 4 (L4) protocol developed by IETF [9]. SCTP is designed to transport Public Switched Telephone Network (PSTN) signaling messages over IP networks. However, the protocol supports a broader range of applications and features. In particular, SCTP supports multihoming, which allows the use of multiple IP addresses for a single association between two SCTP endpoints.

The SCTP association is a broader concept than the TCP connection. During association startup, SCTP provides the means for each SCTP communicating entity to provide the other entity a list of transport addresses (i. e. , multiple IP addresses in combination with an SCTP port) through which that entity can be reached and from which it will originate SCTP packets. These addresses are used as endpoints for different streams. SCTP regards each IP address of its peer as one "transmission path" towards that endpoint. The association spans transfers over all of the possible source-destination combinations that may be generated from each endpoint's list. One of the combinations is selected as initial primary path. If the primary path is considered unreliable, then the packets can be retransmitted on a backup path. Also, the primary path can be replaced with one of the backup paths [4].

Integrated multistreaming and multihoming are the main advantages of SCTP. An important drawback is that applications must be developed with specific support for SCTP (i. e. , the applications must use SCTP sockets) [10] . Consequently, old applications cannot use SCTP unless they are modified. Simultaneous rehoming is possible only for the set of addresses negotiated during association initialization.

C. Combined Multihoming-Mobility Management

Our study contains two protocols that support both mobility and multihoming. The first one, HIP, integrates both features. The second one, Multiple Care-of Address Registration (MCoA), is a multihoming extension for MIPv6.

The HIP specification introduces a new name space, the host identity name space or HIP layer, located between L3 and L4 in the TCP/IP stack [11][12]. The purpose of the HIP layer is to provide a mapping between host identifiers and IP addresses. A host identifier is the public cryptographic key from a public/private key pair that is used to uniquely identify a node. From an operational point of view it is more convenient to work with the hash of the host identifier, which is called a Host Identity Tag (HIT). Applications use HITs to open sockets to and communicate with other hosts. The IP addresses, (i. e. , the locators) are used only for routing purposes. The set of IP addresses associated with a HIT can change over time, for example due to L3 handover or rehoming. These changes are transparent to the applications above the HIP layer.

The advantages associated with HIP include integrated mobility and multihoming, low signaling overhead and transparency to legacy user-level applications [12][13][14]. Furthermore, HIP can handle simultaneous rehoming and simultaneous L3 handover.

HIP's main drawback is the introduction of a new layer to the well-established TCP/IP stack. This requires complex modifications to the operating system running on HIP nodes [8]. Also, in order to support highly mobile nodes, the system requires a rendezvous server (RVS) for location management [15].

Multihoming in MIPv6 can be supported by the MCoA extension, which allows the MN to register multiple CoAs with the HA [4][16]. As a result, the MN can maintain concurrent paths with its CNs by assigning more than one CoA to its network interfaces.

The main advantages of MCoA is that it requires relatively small changes to MIPv6 in order to enable multihoming. One drawback is that the protocol can switch to another CoA only when it detects failures in the communication between the HA and the MN, but is unable to do so for communication between the HA and the CN [17]. Another drawback is that the current specification does not state if multiple addresses can be used at the same time or if one must, for example, choose a single address based on link characteristics [4].

III. Related Work

Magagula et al. [7] discussed handover approaches used by various MIPv6-related mobility management protocols and proposed a handover coordination mechanism based on Proxy Mobile IPv6 (PMIPv6) [18]. The authors used ns-2 simulations to show that their proposed mechanism was more successful than plain PMIPv6 and Mobile IPv6 fast handovers (FMIPv6) [19] in decreasing the handover delay and the packet losses.

Zekri et al. [2] highlighted some of the main technical challenges in providing seamless vertical handover in heterogeneous wireless networks. The article provides a survey on the vertical mobility management process and mainly focuses on decision-making mechanisms. The authors also point out the main research trends and challenges, such as enhancing network availability and QoS, green networking, and solutions for healthcare applications. The main challenges discussed deal with the coexistence of heterogeneous wireless networks.

A comprehensive survey of protocols supporting end-host as well as site multihoming can be found in [4]. The evaluation of multihoming solutions provided there is based on the degree of fulfillment of multihoming goals (i. e. , resilience,

 ISBN: 978-1-61208-318-6

ubiquity, load sharing, and flow distribution). The authors did not explicitly point out the best or worst protocols in terms of performance, but instead they illustrated that each protocol comes with its own advantages and drawbacks. Additionally, they argued that an efficient multihoming protocol cannot be coupled with a single layer, but instead it must be the result of cooperation between multiple layers that act in a concerted manner to meet the same goals. From an end-site perspective, multihoming proposals should not focus only on routing scalability. Instead, they should incorporate native support for the diverse multihoming goals rather than relying on extensions.

In [6], Jokela et al. compared the handover performance of MIPv6 and HIP in a heterogeneous IPv6 network environment. They configured a network environment consisting of a wireless 802.11b network as well as a GPRS network. In their experiment, the MN received a stream of TCP data from a server while performing handover between the two networks. Their measurement results show that the recovery time was 8.05 s for MIPv6 and 2.46 s for HIP.

Ratola et al. [8] compared MIPv6, HIP, and SCTP in terms of architecture, security, and known problems. The purpose of their comparison was to determine which layer (L3, L3.5, or L5) would be best suited for mobility. Based on their comparison, the authors suggest that mobility should be implemented in a new layer between the network and transport layers. In this respect, HIP seems to be a good L3.5 solution for mobility that solves several security, mobility, and multihoming issues at the same time.

Dhraief et al. [20] proposed a novel framework, called MIPSHIM6, that combines SHIM6 [21] and MIPv6, in order to enable both host mobility and host multihoming. In MIPSHIM6, the mobility management is delegated to MIPv6 and the multihoming management to SHIM6. The authors evaluated this framework on a real testbed. They setup an experiment where a MN boots up in a foreign network, binds with its HA and initiates a secure copy (scp) session with a CN. During the next step, the MN establishes a SHIM6 context with the CN . The authors arranged for a HA failure to occur 60 s after the scp session was started. At that point the MN rehomes to the path defined by the SHIM6 context. Unfortunately, there is no data in the paper to indicate how well this solution performs in terms of rehoming time. The TCP throughput plot shown in the paper indicates that it takes 10–15 s for the TCP throughput to increase to the level before the HA failure.

IV. Testbed and Simulation Scenarios

Our performance study was conducted under the OMNeT++ simulation environment. OMNeT++ is a modular, discrete-event simulation framework based on the C++ programming language [22]. It can be used for modeling wired and wireless communication networks, protocols, multiprocessors, distributed or parallel systems, queuing networks and for validating hardware architectures. OMNeT++ is open-source, and it can be used either under the GNU General Public License or under its own license that also makes the software free for non-profit use [23].

We have chosen OMNeT++ because it has an extensive array of modules required by our study, such as HIPSim++ [24] for HIP, the SCTP module [25] from the INET framework, MCoA++ [26] for MCoA and xMIPv6 [27] for MIPv6. The mobility and multihoming features supported by each protocol are summarized in Table I. Note that mobility support in MCoA++ is provided through the xMIPv6 module.

TABLE I: Multihoming and mobility management protocols

Protocol	Mobility	Multihoming
SCTP	No	Yes
MIPv6	Yes	No
MCoA	Yes	Yes
HIP	Yes	Yes

To evaluate the performance of the protocols described in Section II we designed five simulation scenarios: two for mobility and the remaining three for multihoming. The mobility scenarios investigate the handover latency experienced in the case when the MN is using MIPv6 and HIP, respectively. The multihoming scenarios investigate the rehoming time when the host is using HIP, MCoA, and SCTP, respectively.

A. Mobility Scenario for MIPv6

The simulated network topology for this scenario is shown in Figure 1. The rectangle in the background depicts a 850 m by 850 m movement area available for mobile nodes.

The home access point `AP_Home` is connected to the router `Home_Agent` that plays the role of the home agent. Together, they define the home network. The foreign network consists of the foreign access point `AP_1` that is attached to the router `R_1` acting as foreign agent. The coverage areas for `AP_Home` and `AP_1` are overlapped at the boundaries to allow for continuous wireless connectivity. There is approximately 300 m between `AP_Home` and `AP_1`. The bit rate for the backbone links connecting `R_2` to `Home_Agent` and `AP_1` to `R_1` is configured to 1 Gbps. The links between the access points and respective routers are configured as 100 Mbps Ethernet.

The MN is programmed to move from its home network to the foreign network in a straight line at a speed of 1 m/s, resembling a moving pedestrian scenario. The router advertisement (RA) message interval is set to a random number in the range 0.03–0.07 s.

During the simulation, the CN sends every 50 ms a ping packet (i. e. , a ICMP echo message) to the MN. The MN replies to each ping with a ICMP echo reply message. This represents background traffic.

We have configured all MIPv6 nodes to use route optimization, thus avoiding to forward traffic through `Home_Agent`.

B. Mobility Scenario for HIP

The simulated network topology for this scenario is shown in Figure 2. The topology is identical to the one described in Section IV-A with the exception of two additional nodes: the RVS host and the DNS server denoted by `rvs` and `dnssrv`, respectively. The RVS host is a HIP node that allows the MNs to store their actual HIT-to-IP address associations and to make them available to potential communication partners. The DNS server resolves domain names to HITs and IP addresses and also provides RVS information for mobile HIP hosts.

Similar to the MIPv6 scenario, the MN moves from the home network to the foreign network at a constant speed of

 ISBN: 978-1-61208-318-6

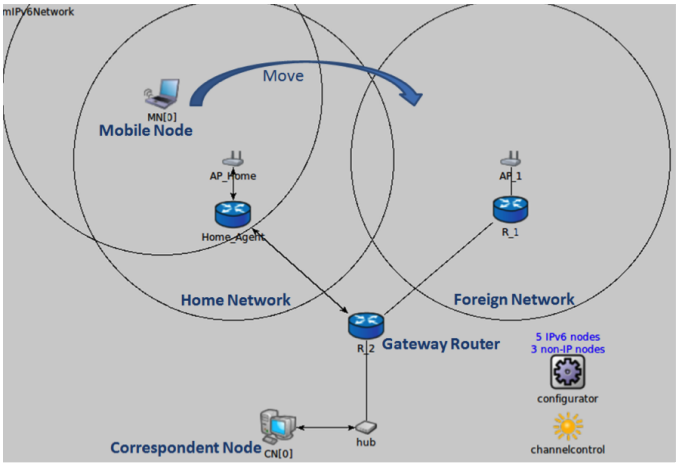


Figure 1: Simulation environment for MIPv6 and MCoA scenarios

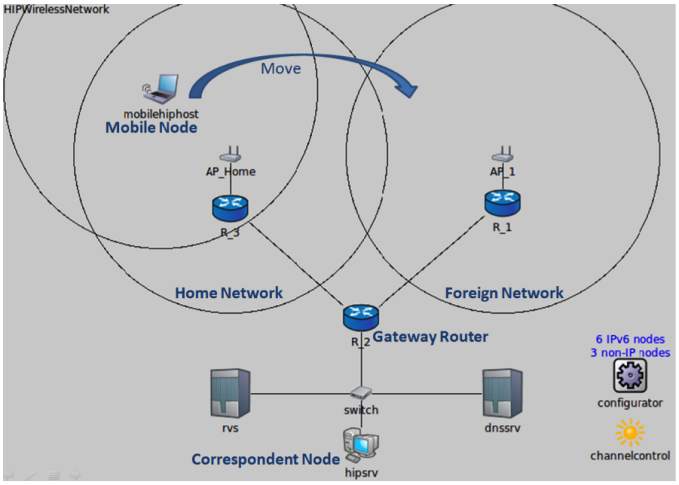


Figure 2: Simulation environment for HIP scenarios

1 m/s. At the start of the simulation, a HIP association is established between the MN and CN. After the association is successfully established, an IPSec Security Association pair is created between MN and CN. At this point, the MN starts to send to the stationary CN (*hipsrv*) one UDP ping request every 50 ms. The reason for using UDP pings is that it was not possible to get the ICMP ping module to work in HIPSim++. We have configured the size of the UDP ping packets to be equal to that of ICMP ping packets and have no reason to suspect any noticeable impact on the simulation results.

MN's movement towards the edge of the home network eventually results in a handover that enables the MN to associate with the foreign access point.

C. Multihoming Scenario for HIP

The topology used here is similar to the one shown in Figure 2, which was used for the HIP mobility scenario. The difference is that in this scenario the MN is equipped with an additional wireless network interface. Our intention is that this scenario should resemble a situation where the MN can

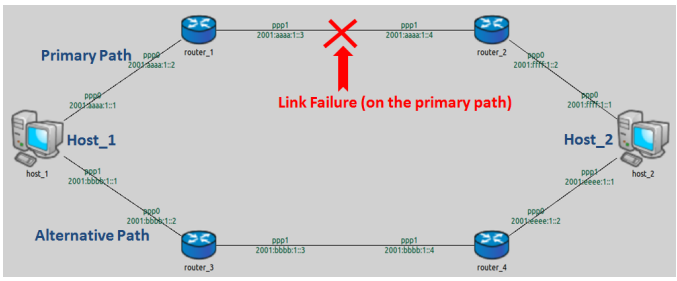


Figure 3: Simulation environment for SCTP

connect to a CN over the (preferred) WiFi interface denoted here by `IF_WiFi` and fallback on the 3G interface, `IF_3G`, when it loses the WiFi connection.

While the MN moves towards the edge of its home network, the signal strength from the home access point `AP_Home` becomes weaker, causing the node to scan for new access points on both its interfaces. We have arranged so that the MN is unable to find another access point over `IF_WiFi`. At some point, the MN will receive the beacon signal from `AP_1` over `IF_3G` and will rehome to the foreign network reachable over the 3G interface.

As explained before, a UDP ping session is established from the MN to the CN. The RA interval is set to a random value between 0.03 s and 0.07 s. The MN is moving away from the home access point at a constant speed of 1 m/s.

D. Multihoming Scenario for MCoA

The MCoA multihoming network topology is similar to the one for MIPv6 shown in Figure 1. The MN is configured to use two interfaces, `IF_WiFi` and `IF_3G`, as explained in the previous section.

The MN is sending every 50 ms an ICMP Echo message to the CN, while moving at speed of 1 m/s.

E. Multihoming Scenario for SCTP

The network topology used to investigate the multihoming capability of SCTP protocol is shown in Figure 3. The SCTP hosts are equipped with two interfaces.

Because SCTP does not have mobility support and also because its mobile extension, mSCTP [28], is not yet available for OMNeT++, we have configured all nodes to use wired interfaces. Therefore, we only tested the multihoming ability of SCTP for data transfer between two stationary hosts.

The links between the routers, `router_1`, `router_2`, `router_3`, and `router_4` form the core network. All these links are configured for 1 Gbps data rate. The Ethernet interfaces of the hosts are configured to use a data rate of 100 Mbps.

The simulation setup aims to study the fault tolerance feature of SCTP when a link on the primary path fails at a random time during data exchange between the hosts. We configured `Host_1` to transfer 10 MB of data to `Host_2`, such that `Host_1` acts as the client and `Host_2` as the server. The size of the transfered data is a tradeoff between keeping the simulation time short and having a long time range where the failure event can occur.

 ISBN: 978-1-61208-318-6

After the endpoints establish a SCTP association, the path consisting of routers `router_1` and `router_2`, is designated as the primary path and the path using the remaining routers, `router_3` and `router_4`, is designated as the alternative path. We have arranged for a link failure to occur on the primary path between `router_1` and `router_2`. The failure event occurs at a time drawn from a uniform distribution between 5.3 s and 7.5 s. This range is well within the time window required to transfer the 10 MB file. The link failure is detected by SCTP, which then redirects the communication through the alternative path.

V. Simulation Metrics

In terms of performance metrics, we measured the rehoming time for multihoming protocols and handover latency for mobility protocols, respectively. For HIP, which supports both multihoming and mobility [13], we collected statistics for both metrics.

A. Handover latency

We define the *MIPv6 handover latency* as the elapsed time between the moment when the MN disassociates from the old access point and the instant when the MN receives the binding acknowledgement (BA) message from the CN [3]. The BA message is sent from the CN to the MN when route optimization is used. Its purpose is to confirm registration of the new CoA. This metric is composed of the following delay components: L2 handover, router discovery, duplicate address detection, home registration, return routability, and correspondent node registration [5][27].

In the case of HIP, every time a HIP-enabled MN changes address it notifies the CN through a sequence of three UPDATE messages. Thus, we define the *HIP handover latency* as the time elapsed from the moment when the MN disassociates from the old access point and until the MN sends out the third UPDATE packet while connected to the new access point [3]. The latency consists of the following delay components: L2 handover, router discovery, duplicate address detection, and peer notification of IP address change (by exchanging three UPDATE messages with the CN).

B. Rehoming time

For a multihomed mobile node with two interfaces, the onset of rehoming is triggered when the MN comes out of range of the old access point, which is connected through the first interface. We therefore define the *HIP rehoming time* as the interval from the moment when the MN starts scanning for the new access point on the second interface until it sends the third UPDATE packet while connected to it. The HIP rehoming time computed this way includes the delays due to L2 handover, configuration of new IPv6 address and HIP update message procedures.

In MCoA++, multihomed MNs use both interfaces simultaneously. When the MN detects a signal from a new access point, it immediately sets up a connection with the new access point via the second interface, which is assigned a new IPv6 address. In our simulation scenario, the ICMP echo requests are sent over the MN's old interface and the replies are received over the new interface. We define the *MCoA rehoming time* as the time elapsed from the moment the MN starts scanning for the new access point over the second interface to the time when the MN receives the BA. The MCoA rehoming time consists of L2 handover, configuration the new of IPv6 address and MCoA signaling procedures.

The *SCTP rehoming time* is defined as the time elapsed between the instant when a failure occurs on the primary path until the moment when data is exchanged on the alternative path.

TABLE II: Performance results with 95% confidence interval

Protocol	Handover latency (s)	Rehoming time (s)
SCTP	N/A	0.99 ± 0.0100
MIPv6	3.32 ± 0.0880	N/A
MCoA	3.32 ± 0.0880	1.66 ± 0.0909
HIP	2.27 ± 0.0922	0.41 ± 0.0004

VI. Performance Results and Analysis

In this section, we present the performance results from our results and share some reflections related to them. Table II shows the statistical mean values for handover latency and rehoming observed in our simulations. We provide also the corresponding 95% confidence intervals. Note that MCoA is a multihoming extension of MIPv6, where the mobility performance of MCoA is similar to that of MIPv6.

Looking at simulation results from the mobility scenarios we can observe that the HIP protocol has an average handover latency of 2.27 s compared to 3.32 s for MIPv6. The higher latency for MIPv6 can be explained by its long signaling phase. In our experiments, MIPv6 required 1.038 s to complete signaling, which is 1 s higher than the time required for HIP signaling. HIP signaling consists of only 3 UPDATE messages exchanged between the MN and CN. In contrast, MIPv6 signaling requires 8 messages, some exchanged between the MN and HA and some between the MN and CN

For multihoming protocols, the results indicate that HIP again has the best performance in terms of the lowest average rehoming time of 413 ms. This is less than half of the SCTP rehoming time (992 ms) and almost a quarter of the MCoA rehoming time (1656 ms). The main reason behind the high performance shown by HIP is proactive IPv6 address configuration. This means that the MN establishes and configures a new IPv6 address on `IF_3G` before it breaks the connection to the home network, hence, performing a soft handover (make-before-brake). When rehoming takes place, it does only a L2 handover followed by HIP signaling, resulting in a very low latency. On the other hand, the rehoming time for MCoA includes L2 handover delay, IPv6 address configuration delay and MCoA signaling latency. The address configuration delay is in fact the largest contributor to the MCoA rehoming time.

VII. Conclusion and Future Work

Our simulation results indicate that HIP has the best performance in both the multihoming and the mobility scenarios. The main reason is HIP's low signaling overhead during handovers and rehoming events. Moreover, HIP implements soft handovers during rehoming events, which decreases the rehoming time by a large factor.

We think that the results presented in this paper indicate that HIP is a suitable component for providing seamless connectivity to mobile and multihomed nodes.

 ISBN: 978-1-61208-318-6

Our future work in the short term will focus on developing missing simulation models for OMNeT++, for protocols such as SHIM6 [4] and mSCTP. This will allow us to extend our current work into a more complete performance analysis of mobility and multihoming protocols. For the longer term, we look forward towards improving the performance of the existing multihoming and mobility protocols.

ACKNOWLEDGMENT

Special thanks go to Dr. Kostas Pentikousis and Mr. Bruno Sousa for their assistance and technical help they provided during our simulations with MCoA.

REFERENCES

[1] E. Gustafsson and A. Jonsson, "Always best connected," IEEE Wireless Communications Magazine, Feb. 2003, pp. 49–55.

[2] M. Zekri, B. Jouaber, and D. Zeghlache, "A review on mobility management and vertical handover solutions over heterogeneous wireless networks," Computer Communications, vol. 35, no. 17, Oct. 2012, pp. 2055–2068.

[3] C. Mugga and D. Sun, "A solution combining both multihoming and mobility in IPv6 heterogeneous environment," Master's thesis, Blekinge Institute of Technology (BTH), Karlskrona, Sweden, Sep. 2013, MEE: 10035.

[4] B. Sousa, K. Pentikousis, and M. Curado, "Multihoming management for future networks," Mobile Networks and Applications, vol. 16, no. 4, Aug. 2011, pp. 505–517.

[5] C. E. Perkins, D. B. Johnson, and J. Arkko, RFC 6275: Mobility Support in IPv6, IETF, Jul. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6275 [retrieved: Dec., 2013]

[6] P. Jokela, T. Rinta-aho, T. Jokikyyny, J. Wall, M. Kuparinen, H. Mahkonen, J. Meln, T. Kauppinen, and J. Kauppinen, "Handover performance with HIP and MIPv6," in Proceedings of Wireless Communication Systems, Mauritius, Sep. 2004, pp. 324–328.

[7] L. A. Magagula, H. A. Chan, and O. E. Falowo, "Handover approaches for seamless mobility management in next generation wireless networks," Wireless Communications and Mobile Computing, vol. 12, no. 16, Nov. 2012, pp. 1414–1428.

[8] M. Ratola, "Which layer for mobility? - comparing mobile IPv6, HIP and SCTP," in HUT T-110-551 Seminar on Internetworking. Sjökulla, Finland: Helsinki University of Technology, Apr. 2004.

[9] R. R. Stewart, RFC 4960: Stream Control Transmission Protocol, IETF, Sep. 2007. [Online]. Available: http://tools.ietf.org/html/rfc4960 [retrieved: Dec., 2013]

[10] A. Dhraief, T. Ropitault, and N. Montavont, "Mobility and multihoming management and strategies," in 14th Eunice Open European Summer School, Brest, France, Sep. 2008.

[11] R. Moskowitz, P. Nikander, P. Jokela, and T. R. Henderson, RFC 5201: Host Identity Protocol, IETF, Apr. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5201 [retrieved: Dec., 2013]

[12] P. Nikander, A. Gurtov, and T. R. Henderson, "Host identity protocol (HIP): Connectivity, mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks," IEEE Communications Surveys and Tutorials, vol. 12, no. 2, Second Quarter 2010 2010, pp. 186–204.

[13] P. Nikander, T. R. Henderson, C. Vogt, and J. Arkko, RFC 5206: HIP Mobility and Multihoming, IETF, Apr. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5206 [retrieved: Dec., 2013]

[14] T. R. Henderson, P. Nikander, and M. Komu, RFC 5338: Using the Host Identity Protocol with Legacy Applications, IETF, Sep. 2008. [Online]. Available: https://tools.ietf.org/html/rfc5338 [retrieved: Dec., 2013]

[15] J. Laganier and L. Eggert, RFC 5204: Host Identity Protocol (HIP) Rendezvous Extension, IETF, Apr. 2008. [Online]. Available: http://tools.ietf.org/html/rfc5204 [retrieved: Dec., 2013]

[16] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami, RFC 5648: Multiple Care-of Addresses Registration, IETF, Oct. 2009. [Online]. Available: http://tools.ietf.org/html/rfc5648 [retrieved: Dec, 2013]

[17] A. Dhraief and A. Belghith, "Suitability analysis of mobility and multihoming unification," in Proceedings of ICWUS, Sousse, Tunisia, Oct. 2010, pp. 1–6.

[18] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, RFC 5213: Proxy Mobile IPv6, IETF, Aug. 2008. [Online]. Available: http:// tools.ietf.org/html/rfc5213 [retrieved: Dec., 2013]

[19] R. Koodli, RFC 5568: Mobile IPv6 Fast Handovers, IETF, Jul. 2009. [Online]. Available: http://tools.ietf.org/search/rfc5568 [retrieved: {Dec., 2013}]

[20] A. Dhraief, I. Mabrouki, and A. Belghith, "A service-oriented framework for mobility and multihoming support," in Proceedings of MELECON, Medina Yasmine Hammamet, Tunisa, Mar. 2012, pp. 489–493.

[21] A. García-Martnez, M. Bagnulo, and l. van Beijnum, "The shim6 architecture for ipv6 multihoming," IEEE Communications Magazine, vol. 48, no. 9, Sep. 2010, pp. 152–157.

[22] A. Varga and R. Hornig. OMNeT++ community site. [Online]. Available: http://www.omnetpp.org [retrieved: Dec., 2013]

[23] ——, "An overview of the OMNeT++ simulation environment," in Proceedings of Simutools, Marseille, France, Mar. 2008, pp. 1–10.

[24] L. Bokor. HIPSim++: A host identity protocol (HIP) simulation framework for INET/OMNeT++. [Online]. Available: http://www.ict-optimix.eu/index.php/HIPSim [retrieved: Aug., 2013]

[25] I. Rüngeler, M. Tüxen, and E. P. Rathgeb, "Integration of SCTP in the OMNeT++ simulation environment," in Proceedings of Simutools, Marseille, France, Mar. 2008, pp. 1–8.

[26] B. Sousa, M. Silva, K. Pentikousis, and M. Curado, "A multiple care of address model," in Proceedings of Computers and Communications, Kerkyra, Greece, Jun. 2011, pp. 1–6.

[27] F. Z. Yousaf, C. Bauer, and C. Wietfeld, "An accurate and extensible mobile IPv6 (xMIPV6) simulation model for OMNeT++," in Proceedings of Simutools, Marseille, France, Mar. 2008, pp. 1–8.

[28] S. J. Koh, Q. Xie, and S. D. Park, Mobile SCTP (mSCTP) for IP Handover Support, IETF, Oct. 2005, draft-sjkoh-msctp-01.txt.

 ISBN: 978-1-61208-318-6

Integrating CARMNET System with Public Wireless Networks

Przemyslaw Walkowiak
Institute of Control and Information Engineering
Poznan University of Technology
Poznan, Poland
przemyslaw.walkowiak@put.poznan.pl

Radoslaw Szalski
Institute of Control and Information Engineering
Poznan University of Technology
Poznan, Poland
radoslaw.szalski@put.poznan.pl

Salvatore Vanini
Institute for Information Systems and Networking
The University of Applied Sciences and Arts of Southern Switzerland
Manno, Switzerland
salvatore.vanini@supsi.ch

Armin Walt
Administrator of Lugano WiFi
wlan-partner.com AG
Zurich, Switzerland
armin.walt@wlan-partner.com

***Abstract*—In this paper, we present two scenarios of the CARMNET system application in wireless networks. The CARMNET system is an advanced experimental resource allocation framework based on the Network Utility Maximization model that may be integrated with a standard telecom operator infrastructure. Having developed the CARMNET system, we tested it in a publicly available wireless network, operated by the city of Lugano, Switzerland. In the presented scenarios, the CARMNET was employed to extend the network coverage and to improve it by providing the feature of seamless handover. The results indicate the ability of successful integration of the CARMNET system with an existing real-world network.**

***Keywords*-Wireless Mesh Networks; Network Utility Maximisation; network scalability; DANUMS; seamless handover**

I. INTRODUCTION

The main idea behind the *Carrier-grade delay-aware resource management for wireless multi-hop/mesh networks* (CARMNET) [1] system is to make the user-provided Internet access a viable alternative to the currently widespread 3G/4G-based mobile Internet access. Simultaneously, the CARMNET system introduces an advanced, distributed resource management mechanisms integrated with a telecom operator's IP Multimedia Subsystem (IMS) and Authentication, Authorisation and Accounting (AAA) infrastructure and provides support for generalized network node mobility. Put into practice, these ideas allow us to improve existing, wireless networks with features like transparent extension of range and better network resource utilisation.

A. Drawbacks of traditional wireless networks

Traditional networks, including wireless ones, require hardware investments in order to scale. To support more users and provide a larger network range, wireless access points must be either upgraded, or their number simply has to be increased. The issue is even more noticeable the higher the number of users and the wider the required coverage. However, the cost of new hardware is not the only one. There is also the price of additional service and management. The proposed system shifts the costs of the hardware towards the users of the network, giving in return benefits to those users who share their network access. Another issue of standard wireless networks is that they seldom incorporate any resource management system, the lack of which may lead to degradation of user experience. This is particularly evident in cases when many users, with different delay and throughput requirements, are connected to the network at the same time. For example, if one of them establishes a Voice over IP (VoIP) connection (delay sensitive flow) and the other downloads a large file via File Transfer Protocol (FTP) (throughput demanding flow), the characteristic of FTP protocol will cause the decrease of the VoIP call quality and dissatisfaction of its recipients. Most of Internet providers deal with the problem by limiting the available throughput to the user or by blocking unwanted services. However, the CARMNET system, thanks to the adoption of the Delay-aware Network Utility Maximisation (DANUM) [2] framework and the introduction of applications' and users' specific profiles, can serve heterogeneous traffic according to its requirements [3]. This allows the delay of VoIP call to be minimized and the throughput of file transmission to be maximized, while providing a high level of user experience in both cases. Finally, the extension of the network's range can be achieved using a mobile node sharing its Internet access with remote nodes. The switch should be done transparently, ensuring the correct routing of packets without degradation of QoS perceived by the users. Thanks to the integration with the WiOptiMo framework [4], the CARMNET system is able to manage this scenario as well.

In this paper, we describe two scenarios where the CARMNET system was integrated with the Lugano WiFi network infrastructure. Each scenario is presented with a description of a real-world use case of the CARMNET architecture and its advantages. Issues regarding the integration of the CARMNET system into an existing wireless networks are presented in Section IV. The CARMNET system itself is thoroughly described in Section III. To validate our system

 ISBN: 978-1-61208-318-6

integration, we executed field tests, since this approach allows to uncover most of the problems and it is also more adequate from integration perspective. Lugano, Switzerland, was chosen for our tests. It is a city of $65,015$ inhabitants and an area of $75.81\ km^2$. Free WiFi access is available for tourists in the city centre. Details about the configuration of the Lugano WiFi network are presented in Section V. To test the integration outcome, we carried out two experiments (described in Section VII), whose results are presented in Section VIII. It is important to highlight that although the experiments were performed in Lugano, the CARMNET system can be easily integrated with any wireless WiFi infrastructure.

II. Related Work

Many resource allocation systems based on the Network Utility Maximisation (NUM) model exist [5]–[7] and determine the utility of flows according to their measured properties. However, only a few of them were implemented and tested in a real wireless mesh network [5], [6]. It is an important step in an attempt to integrate proposed solutions with public networks. Unfortunately, such approaches were not sufficient to effectively measure the utility of both delay-sensitive and throughput-oriented flows. The system proposed in [2](DANUM System (DANUMS)) takes both parameters into consideration. Moreover, DANUMS has a well-tested implementation, which allows researchers to focus on specific parts of the CARMNET system.

Network mobility support is addressed in RFC 4886 [8] and in several architectural solutions presented in literature. SyncScan [9] is a Layer-2 procedure for intra-domain (between Access Points (APs)/same domain) handoff in 802.11 infrastructure mode networks. It achieves good performance at the expense of a required global synchronization of beacon timings between clients and access points. iMesh [10] provides low handoff latency for horizontal (same network technology) Layer-3 handoffs in Wireless Mesh Networks (WMNs). Its main drawback is that handoff delay depends on the number of nodes along the path between the new AP and the old AP. BASH [11], focuses on the design of a horizontal—intra-domain—Layer-2 seamless handoff scheme for 802.11 WMNs. BASH's main drawback is that it requires modifications on the mobile client's side for managing the handoff protocol. The method proposed in [12] uses tunnelling, as it is the case with RFC 4886 and the standard Mobile IPv6 solution [13]. Tunnelling introduces extra delay for the encapsulation/decapsulation of packets and has low flexibility intrinsically. Finally, SMesh [14] provides a 802.11 mesh network architecture for both intra-domain and inter-domain handoffs (between Internet-connected APs/different domains). For intra-domain handovers, SMesh generates high network overhead, which grows linearly with the number of clients. In case of inter-domain handovers, network overhead is directly proportional to the number of connections each client has.

III. CARMNET System

A. Integration of WMN and IMS

The integration of WMNs and IMS infrastructure provides many benefits for both users and telecom operators. WMNs enable extended network coverage without the need to expand the static infrastructure. At the same time, telecom operators may easily manage users and offer them additional services. In the presented solution, Session Initiation Protocol (SIP) is used as an integration protocol. Each node acts as a SIP User Agent (SIP UA). SIP, however, is designed to control access to the mesh network only. The IMS infrastructure is used for authentication, authorization, storage of user profile information as well as for accounting and charging. The registration is performed by the SIP UA implementing the authentication/authorization functions of IMS AAA by means of the standard SIP REGISTER message. The SIP SUBSCRIBE/NOTIFY mechanism has been applied to transport the user-related information.

B. The Architecture of the System

The architecture of the CARMNET system [1], [15] consists of multiple components located both on the client- and server-side (see Figure 1).

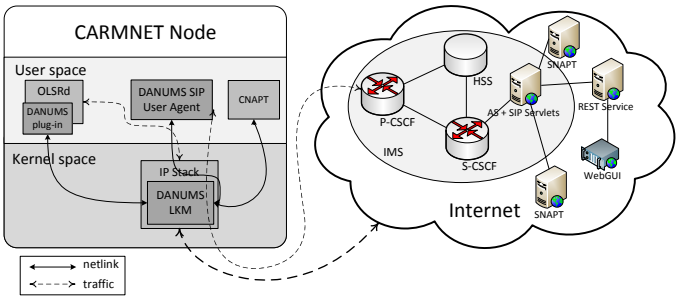


Figure 1: Interactions between CARMNET components.

DANUMS Loadable Kernel Module (LKM) [2] is an implementation of the DANUM model developed for the Linux environment. This subsystem works at the kernel level, which allows for a deep integration in the networking stack, necessary to introduce custom queuing and scheduling subsystems. For the network path resolution, the Optimised Link State Routing Protocol daemon (OLSRd) — a popular implementation of the Optimised Link State Routing Protocol (OLSR) protocol — is used. Since OLSR messages' additional purpose is to distribute data needed by the packet scheduler, a special way of communication between user-space (OLSRd) and kernel-space (DANUM LKM) had to be applied. The Netlink protocol serves that role. Network mobility support is performed by the CARMNET mobility subsystem, which is based on the WiOptiMo framework. A client proxy (CNAPT) installed on any CARMNET wireless node provides mobility services to users who subscribed for them. It intercepts traffic flows associated to the mobility service and relays them to a server proxy (SNAPT) according to their requirements in terms of bandwidth and delay, in order to provide the desired Quality of Service (QoS). Multiple SNAPTs are located on the Internet to manage scalability and avoid concentrating traffic flows in a single spot. The WiOptiMo framework is also used for providing the DANUMS LKM with throughput and delay measurements for flows endpoints that are beyond the CARMNET network. This is done through *procfs* interface [16]. The direct communication of DANUM LKM located at the client's node with the IMS infrastructure located in a network is a difficult task. A high-level SIP protocol in the low-level

 ISBN: 978-1-61208-318-6

kernel module has to be implemented, in a way that preserves the high efficiency of the Linux kernel. Therefore, we have developed a SIP UA in the user-space running on the client side, which is responsible for asynchronous communication between LKM and IMS. The communication between LKM and SIP UA is realized using the Netlink protocol, whereas the communication between SIP UA and IMS is realized using the custom CARMNET protocol, based on XML encapsulated in the standard SIP messages. This allows the design to be further compatible with the IMS architecture. The Web User Interface (WebGUI) is a WWW application dedicated to users of the CARMNET network. Users are allowed to configure their own profiles and to bind the utility function (and its arguments) to the type of traffic (e.g., WWW, e-mail, Skype). Moreover, WebGUI shows the utility unit account balance and reports about transmitted traffic and its 'price' in virtual currency. The concept of virtual currency (Denarii) is described in detail in Subsection VI-C. Two types of servlets are located on the application server. The first one is responsible for managing AAA functions and user profiles in the CARMNET network. After connecting to the network, each client has to be authenticated and authorized in the SIP servlet before starting an Internet session. During the session, each node in the network reports to the SIP servlet the amount and the type of traffic it has served. All CARMNET-specific information about users is accessible through the second servlet. As the information transport protocol, the servlet implements REST-like service and WebGUI acts as a REST client. To be part of the CARMNET network, users must install the abovementioned, client-side software, on their devices.

IV. Integration with Existing Networks

In general, there are two different approaches to integrating systems, such as CARMNET, into 3rd party, already established network solutions. The first one limits the integration to the network clients only, but reuses existing network's AAA system at the same time. This approach is aimed mainly at the extension of range of an existing wireless network, so that it covers the previously inaccessible areas, without the use of any additional static infrastructure. This is achieved by deploying additional software on clients' devices only. This software is responsible for advanced resource management as well as for network access sharing and has to be installed by all the connecting clients. As mentioned above, the optimization of resource usage is done in a distributed way by each of the CARMNET-compatible network components. In this scenario, optimization is performed locally, between clients' nodes and not throughout the network. Access points are not modified. Advantages of the CARMNET system are most visible when the DANUMS-enabled client shares its connection to another client.

The second approach, requires an additional modification to the software of the existing infrastructure, i.e., DANUMS has to be installed on all access points. Besides expanding the range and optimizing local resource usage, the CARMNET system provides a distributed optimization across the entire network. Another requirement of the CARMNET system is the integration with the existing AAA subsystem, which is necessary to allow clients to set up their profiles via the CARMNET WebGUI. This way, a more flexible accounting system based on users' perception of flows utility might be applied. This integration relies on implementation of an additional authorization plug-in for the CARMNET IMS subsystem.

For real-world scenarios, the first approach is easier to integrate. Many already existing network infrastructures have well established AAA capabilities. Integrating with those would require additional implementation effort making the solution less interesting. Moreover, we are currently not prepared to support any other platform than the Linux-based one. Therefore, we have used the first approach, which is less complex and easier to be fully realized. Furthermore, many use cases (including our proposed scenarios) are still viable without full integration.

V. The Lugano WiFi Network Configuration

The Lugano WiFi network is available in different discontinuous areas of Lugano. For our experiments, we have decided to focus only on the city centre area which, due to the way it is covered, is more appropriate for the scenarios presented in this paper.

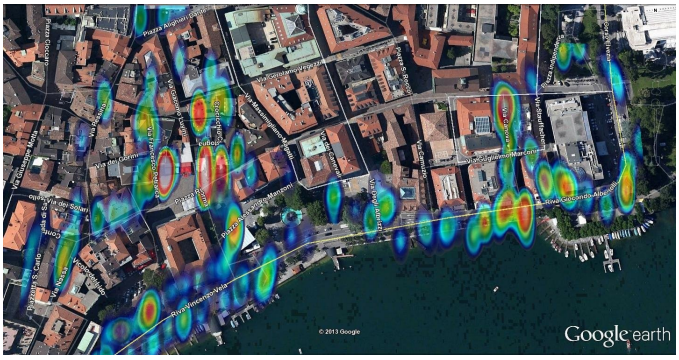


Figure 2: Lugano WiFi coverage in the city centre.

Figure 2 describes the actual configuration of the Lugano WiFi network in the city centre. It was obtained using the *Kismet* layer-2 wireless network detector to eavesdrop on 802.11 beacons and register the Global Positioning System (GPS) coordinates of the points where WiFi fingerprints were collected. In the picture, these points can be seen as circles with different colours, whose gradients vary from red, to yellow, green, and blue. They indicate gradual changes in Received Signal Strength (RSS), from strong to weak respectively.

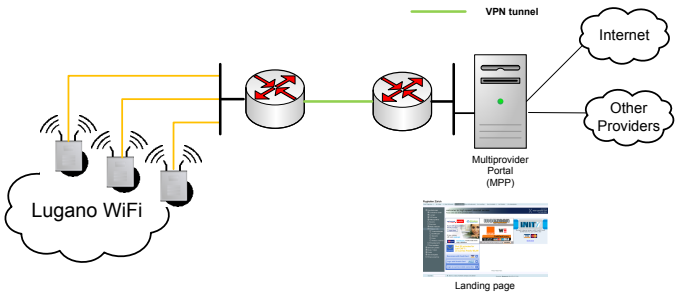


Figure 3: Lugano WiFi architecture.

The network is managed by the WLAN-Partner company [17], which is located in Zurich. The solution currently used

by Lugano WiFi to manage handovers is based on tunnelling (Figure 3). A Virtual Private Network (VPN) tunnel is opened between the core router and the WLAN-Partner data centre. A software called Multiprovider Portal (MPP) performs AAA functions. It is based on the Linux application for packet filtering *iptables* and consists of a combination of rules and rulesets. Those rules mark the network traffic and determine if it belongs to an authorized session, with Internet access capability, or an unauthorized (new) session. In the latter case, traffic is redirected to a landing page. The MPP system allows each mobile client (identified by its Medium Access Control (MAC) address) to keep the same Internet Protocol (IP) address when moving and associating to a different access point. A lease tracker daemon checks for a session timeout, which is set to 30 minutes—after that time is elapsed, the user is redirected to the landing page and has to reconnect to access the network again. During the experiments, a set of special accounts without session time limit were used, for practical reasons. Finally, the average number of devices connected to the network is 1340 per month (about 45 per day), while the number of sessions per workday is 5050 and the average duration is about 25 minutes.

VI. CARMNET Usage Scenarios

A. Network Coverage Extension

Extending the range of static wireless networks requires the deployment of additional access points. This process is often complicated because of power or network infrastructure requirements. The CARMNET system allows for transparent extension of range using not only static infrastructure, but the clients' devices as well. Each user's device is equipped with the CARMNET client software, which enables the sharing of Internet access with other CARMNET clients. The process of sharing is managed by a resource management system, which takes care of the users' preferences in terms of how to serve their incoming and outgoing traffic. Additionally, our system introduces a unified charging system based on the transfer of virtual currency, which enables incorporation of a more complex model of incentives and collaboration enforcement (i.e., access to additional services). The Lugano WiFi network may provide coverage in locations where it previously was not possible. This would allow users to access the Internet even while being outside of the standard infrastructure. What might be especially beneficial to the telecom operators is the potential this scenarios has, to reduce last mile network operational costs. We can imagine a situation in which a single user is sharing Internet access to five other users. From the operator's perspective, the amount of bandwidth used by these six people is limited to the bandwidth accessible to the sharing user only. Another benefit is that each additional sharing user extends the signal range by the range provided by her device. Every network user and local facility, in public WiFi's range, can be a sharing node. Local facilities, since they are usually static, can continuously share the Internet connection.

B. Seamless Horizontal Handover

The CARMNET infrastructure can be used to bridge the gap between two (or more) scopes of WiFi signal. For this solution to be truly usable to the users, we need to provide handover functionality between the CARMNET network and the public WiFi. This way users' ongoing sessions will not be affected as they move from one device's WiFi signal range to the other. We have identified an uncovered area of the Lugano WiFi network and used CARMNET's sharing nodes to set up a simple WiFi network, and provide coverage in the indicated area (see Figure 4). The CARMNET's nodes were behind NAT gateways with different IP addresses. We wanted to test the capability to support WiFi micro-mobility (handover between access points of the same provider). For this purpose, a mobile CARMNET node moved linearly across the coverage area of the other CARMNET's sharing nodes. Using this configuration, we collected measurements about goodput, handoff latency (time to perform a complete handover procedure) and packets loss rate.

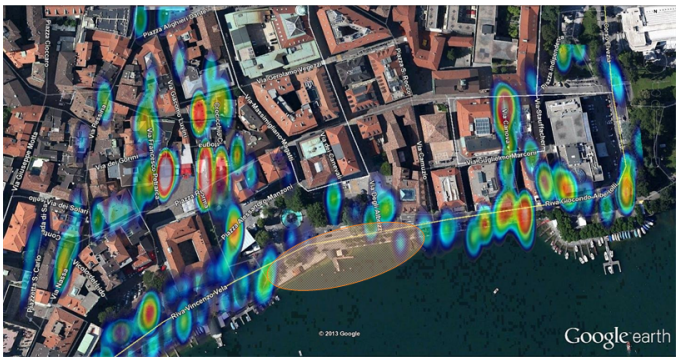


Figure 4: The area not covered by Lugano WiFi, chosen for the handover experiment.

C. The Usage of Virtual Currency

The CARMNET system provides a concept of virtual currency [2] called *Denarii*. It is used internally for flow management, but this is not its only possible use. In one scenario, it could be used for city promotion through discounts and vouchers for various attractions. Tourists may collect Denarii for selfless behaviour and turn them into various rewards, such as museum tickets. Cities could provide many other incentives for users, who would be encouraged to share their Internet connection. This would be beneficial to the whole network and local community.

VII. Experiments

Two experiments have been carried out on site, to verify the effectiveness of integration. Their results are presented in Section VIII. Before performing the experiments in the field, each one has been performed locally in a wireless network testbed [3] developed for CARMNET testing purposes. This testbed allows for an automated scenario execution using a set of ALIX integrated network nodes equipped with two network interfaces (a wired one for commands execution and a wireless one for experiment traffic). Those nodes are managed centrally using the wnPUT2 server capable of booting, restarting and issuing commands to groups as well as individual machines. This testbed helped us validate the following experiments.

 ISBN: 978-1-61208-318-6

A. Network Coverage Extension

An additional wireless CARMNET network, consisting of 3 nodes, has been set up. A stationary node n_1 was connected to Lugano WiFi network, during the whole experiment, and was acting as a final gateway for both nodes n_2 and the *user* node. Node n_2 was located such that it was connected to n_1 but was not in range of the public WiFi network. It had Internet access by means of Internet sharing from node n_1 only. The third node was simulating a user travelling along a linear path, from n_1, towards n_2 and beyond (see Figure 5). The user was initially connected to node n_1. While moving away from n_1, as the signal strength of n_1 lowered, it changed the default gateway to n_2. This was possible thanks to the OLSR daemon monitoring the Expected Transmission Count (ETX) routing metric extension, determining changes in path quality. Simultaneously, the user node lost the signal of public WiFi network. During movement, the user node was issuing Internet Control Message Protocol (ICMP) Echo Requests to an external address, reachable through the Internet, to check connectivity. At the end of the path, a Transmission Control Protocol (TCP) connection was established to download a 15 MB file using the *wget* application.

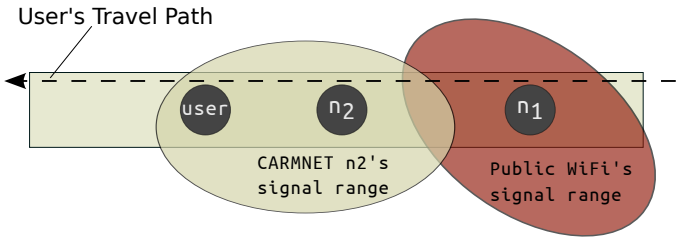


Figure 5: Network coverage extension experiment topology. Note that node n_1's signal range has been omitted for clarity.

B. Seamless Handover

There is a gap between signal ranges of the Public WiFi network at the experiment site. A CARMNET network, consisting of 3 nodes, has been set up to cover the signal gap. Nodes were arranged in a linear topology, so that the first (n_1) and last (n_2) nodes were connected to the Public WiFi network, to fill in the gap. Another, third CARMNET node was simulating a user travelling along the indicated path (see Figure 6). During its movement, it would eventually move out of the Public WiFi's signal #1 range. Since the mobile node was a CARMNET node, once the signal #1 strength decreased under a certain threshold, OLSRd automatically changed the routes and, as expected, the node was connected to the existing CARMNET network. A similar situation had taken place as the user node was leaving CARMNET network's range and approaching Public WiFi's signal #2 range. During the experiment, traffic logs across all of the nodes were collected.

VIII. Results

The first experiment proved a successful extension of the Public WiFi network range, using the CARMNET system. Being connected to the CARMNET network, the user node was still able to access the Internet provided by Lugano WiFi, while being out of range of its infrastructure. No modifications

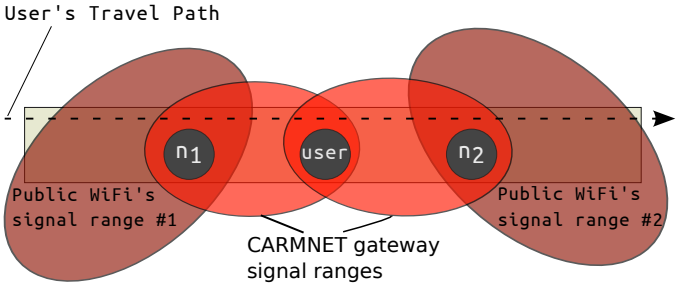


Figure 6: Handover experiment network topology.

to existing infrastructure were necessary. Figure 7 shows the Denarii balance over time for each node participating in the experiment, which indicates how much each node shared/used the Internet connection. User node was initiating all the traffic and thus paying for it: this can be inferred from a constantly dropping Denarii balance. Nodes n_2's virtual unit balance was oscillating around 0, since it was forwarding traffic only. It is worth noting, that ultimately, node n_1 was sharing the Internet connection for the rest of the nodes. It was thus rewarded by the CARMNET system with the highest amount of Denarii.

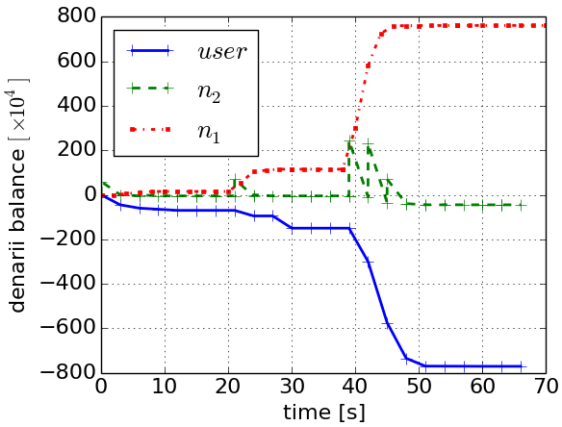


Figure 7: Denarii balance for the three CARMNET nodes during the first experiment.

Figure 8 shows a graph of TCP throughput for each of the CARMNET nodes participating in the second experiment. In the initial phase of the experiment, only node n_1 was in the range of the user node. It was thus chosen as a default gateway and shared Internet access from the Lugano WiFi network. As the user moved away from n_1, its RSS lowered and the default route began to degrade (the value of ETX metric was rising). During this time, all of the traffic was sent through node n_1, which can be seen in Figure 8. Around the middle of the experiment, user node began to detect signal from the second gateway (n_2). Once the alternative route had good enough ETX metric, the CARMNET mobility module based on WiOptiMo had switched routes. This corresponds to a gap in flow's throughput around 103rd second. The connectivity was not interrupted as the flow was forwarded by the second node n_2. The steep, brief increase in throughput, seen on the graph around 105th second, is caused by an accumulation of packets in the DANUMS queue after the handover has

 ISBN: 978-1-61208-318-6

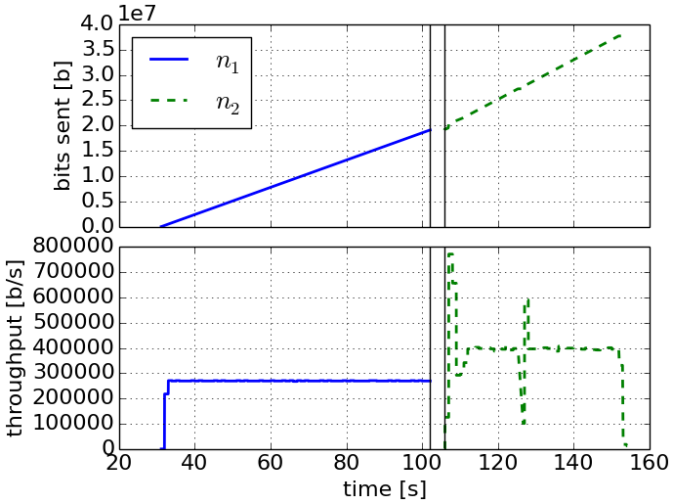


Figure 8: Throughput and total bits sent, registered at the 3 CARMNET nodes during the seamless handover experiment.

been completed. The overall goodput of the flow, measured during the experiment, was 1.01 Mbit/s. It took 3.78 seconds to complete the handover. Another metric is the rate of packet loss, which was around 0.21% during the transmission. From the above measurements, we can conclude that the handover was performed correctly and successfully.

IX. Conclusion and Future Work

We have shown that the CARMNET system can be successfully integrated into existing public wireless networks. Experiments presented in the paper illustrate the two most important real-word use cases of the system. The key factor enabling integration is the system's compatibility with standard AAA mechanisms used by the telecom operators such as the one implemented in the IMS platform. Even without modification of existing software, CARMNET provides many benefits for network users and Internet providers, e.g., network coverage increase, potential to reduce last mile operational costs and seamless handover.

In order to operate inside the CARMNET system, new users need an initial amount of virtual currency, which allow them to pay for their traffic when they have not yet got paid for serving flows themselves. Currently, new users are granted a fixed amount of Denarii. Our future work includes testing what amount of initial virtual currency is appropriate. It cannot be too small, because sharing users would not have a chance to earn enough, which might lead to a deadlock. It cannot be too big, since users would not be compelled to earn Denarii and thus share the Internet access. There is also the risk that users will not be paid for Internet sharing [1]. Our aim would be to minimise this risk by incorporating trust mechanisms akin to those implemented in Peer-to-Peer (P2P) networks.

Acknowledgement

This work is supported by a grant from Switzerland through the Swiss Contribution to the enlarged European Union (PSPB-146/2010, CARMNET).

References

[1] M. Glabowski and A. Szwabe, "Carrier-Grade Internet Access Sharing in Wireless Mesh Networks: the Vision of the CARMNET Project," The Ninth Advanced International Conference on Telecommunications, Jun. 2013.

[2] A. Szwabe, P. Misiorek, and P. Walkowiak, "Delay-Aware NUM system for wireless multi-hop networks," in European Wireless 2011 (EW2011), Vienna, Austria, Apr. 2011, pp. 530–537.

[3] P. Walkowiak, M. Urbanski, M. Poszwa, and R. Szalski, "Flow classification in delay-aware num-oriented wireless mesh networks," in MESH 2013, The Sixth International Conference on Advances in Mesh Networks, 2013, pp. 33–38.

[4] S. Giordano, D. Lenzarini, A. Puiatti, M. Kulig, H. Nguyen, and S. Vanini, "Demonstrating seamless handover of multi-hop networks," in Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, ser. REALMAN '06. New York, NY, USA: ACM, 2006, pp. 128–130. [Online]. Available: http://doi.acm.org/10.1145/1132983.1133011

[5] U. Akyol, M. Andrews, P. Gupta, J. D. Hobby, I. Saniee, and A. Stolyar, "Joint scheduling and congestion control in mobile ad-hoc networks," in The 27th IEEE International Conference on Computer Communications (INFOCOM 2008), Apr 2008, pp. 619–627.

[6] B. Radunović, C. Gkantsidis, D. Gunawardena, and P. Key, "Horizon: Balancing TCP over multiple paths in wireless mesh network," in Proceedings of the 14th ACM international conference on Mobile computing and networking, MobiCom 2008, 2008, pp. 247–258.

[7] M. Neely, "Delay-based network utility maximization," In Proc. IEEE INFOCOM 2010, 2010, pp. 1–9.

[8] T. Ernst and L. H., "Network mobility support goals and requirements," in RFC 4886, July 2007.

[9] I. Ramani and S. Savage, "Syncscan: Practical fast handoff 802.11 infrastructure networks," in 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), vol. 1, 2005, pp. 675–684.

[10] V. Navda, A. Kashyap, and S. R. Das, "Design and evaluation of imesh: an infrastructure-mode wireless mesh network," in IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WOWMOM), Italy, June 2005.

[11] Y. He and D. Perkins, "Bash: A backhaul-aided seamless handoff scheme forwireless mesh networks," in International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2008). IEEE, June 2008.

[12] R. Huang, C. Zhang, , and Y. Fang, "A mobility management scheme for wireless mesh networks," in IEEE GLOBECOM 2007, Washington DC, USA, November 2007.

[13] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in ipv6," in RFC 3775, June 2004.

[14] R. M.-E. Y. Amir, C Danilov and N. Rivera, "The smesh wireless mesh network," ACM Transactions on Computer Systems, vol. 28, no. 3, September 2010.

[15] P. Walkowiak, M. Urbański, A. Figaj, and P. Misiorek, "Integration of danum-based carrier-grade mesh networks and ims infrastructure," in Ad-hoc, Mobile, and Wireless Network. Springer Berlin Heidelberg, 2013, pp. 185–196.

[16] S. Vanini, D. Gallucci, S. Giordano, and A. Szwabe, "A delay-aware num-driven framework with terminal-based mobility support for heterogeneous wireless multi-hop networks," in ICTF 2013 Information and Communication Technology Forum.

[17] "WLAN PARTNER." [Online]. Available: http://www.wlan-partner.com/

 ISBN: 978-1-61208-318-6

Trends in Local Telecommunication Switch Resiliency

Andrew P. Snow
School of Information & Telecommunication Systems
Ohio University
Athens, Ohio, USA
e-mail: asnow@ohio.edu

Gary Weckman
Department of Industrial & Systems Engineering
Ohio University
Athens, Ohio, USA
e-mail: weckmang@ohio.edu

Abstract— **This paper presents a time series analysis of outage causality trends for local telecommunication switches in the United States. Additionally, the resiliency of local switches is assessed by examining changes in severity of outages over time by causality. Almost 13,000 Public Switched Telephone Network (PSTN) switch outages are examined over a 14-year period. Causality trends were examined from both resiliency and reliability perspectives, for scheduled outages, and failure induced cause categories such as procedural errors, design errors, random hardware failures, and external events. Examples of reliability growth, constancy, and deterioration are noted among these casual categories. Likewise, examples of increasing, constant and decreasing impact trends are also noted. To examine resiliency, a novel severity index metric is introduced that is not only intuitive, but also robust, given the long tailed distribution of outage impact. The new index allows time series comparison between causality reliability and outage impact trends. Interestingly, in some instances causality reliability trends were different from the corresponding impact trends. For example, when all outage causes are combined, a reliability growth trend is indicated while outage impact is constant. To get a good perspective on resiliency, both reliability and outage impact trends must be examined. Trends are assessed both graphically and analytically, and conclusions are reached with strong statistical inference.**

Keywords- reliability growth; resiliency, outage index; homogeneous poisson process (HPP), non-homogeneous poisson process (NHPP), Laplace trend test, time series of events, fault management.

I. INTRODUCTION

Telecommunication switches are an important subsystem of the Public Switched Telephone Network (PSTN). Along with the transmission and signaling subsystems, these switches provide end to end connections to subscribers. Although the PSTN in the U.S. is used predominantly for landline voice services, many wireless calls use many of the same facilities, especially for regional calls. Additionally, PSTN and wireless switches are often different models of the same switch vendor product line, as the switching functions are very similar. So wireline switches, besides serving millions of subscribers and deserving of study in their own right, are also good proxies for wireless mobile switching centers [1]. The PSTN is certainly migrating to voice over internet protocol (VoIP), but this migration will take many years, and local switches are likely remain in service for years to come [2]. At the beginning of 2011, there were 117 million subscribers connected over local loops to circuit switched local switches, and 32 million VoIP subscribers [3].

Additionally, local switches are access nodes, and access nodes, be they in a circuit or a packet switch networks, are very important as they represent the gateway to network services for users. For instance, the methods presented in this paper, and the types of insights that can be gleaned from failure and outage data are as relevant for local PSTN switches, as they are for Internet Service Provider (ISP) access nodes.

Trends are important in the dependability of systems, subsystems, and components. Statisticians identify trends – engineers endeavor to embrace favorable trends and influence for the better negative trends. A key to understanding reliability trends is to recognize and identify the hazards in which the item, system or service operates. Additionally, management must make reliability programs a priority [4]. In order to change a trend, we look for approaches offering insights into why failures are occurring. Telecommunication switch reliability results from complex interaction between software, hardware, operators, traffic load, and many environmental factors. By knowing failure causes and trends, switch vendors and service providers may take actions to change trends. Barnard argues that reliability engineering must endeavor to continually improve systems and products before and during the operational phase, using such techniques as the Failure Reporting, Analysis and Corrective Action System (FRACAS) [5]. In network management, this is a fault management function.

Additionally, all outages are not equal, as some might affect hundreds for long periods, while others might affect hundreds of thousands for short periods. Outages represent resiliency deficits, and trends in resiliency are as important as trends in reliability. This paper endeavors to examine reliability and resiliency trends, as they relate to outage causality. In this way, we can not only arrive at an overall assessment, but also assessments segregated by cause.

This study uses local switch outage data reported to the Federal Communications Commission (FCC) representing individual switch outage incidents of at least two minutes in duration from 1996 through 2009, collected from [6]. Unfortunately, after 2009, the FCC no longer required carriers to report this data. As each reported switch outage includes date, time, and outage cause, a time series by cause was created by us to assess causality trends. In addition, as the reports also included the size of each switch out (in

 ISBN: 978-1-61208-318-6

access lines) and duration, time series outage impact trends could also be assessed. The reporting Carrier classified each incident using one of fifteen different cause codes. We further aggregated the fifteen causes into six outage causality categories. Causality trends are of paramount interest if the objective is to understand failure and outage patterns. This way, remedial action plans can be taken by network operators to improve network performance.

Section II presents a literature review of outage impact, local switch reliability, and reliability of repairable systems. Section III introduces the specific research questions being addressed in this paper while Section IV overviews the time series methods used to assess local switch reliability and resiliency, including a new resiliency metric. Section V presents the causality trends in graphical and tabular form, including reliability ad resiliency comparisons. Section VI summarizes conclusions while Section VII addresses research limitations and suggests future research directions.

II. LITERATURE REVIEW

A. Outage Impact

A system is resilient when it has "the capability of a system to maintain its functions and structure in the face of internal and external change and to degrade gracefully when it must." [7]. The role of modeling in understanding and promoting the resilience of critical infrastructures, including wireline and wireless telecommunications, has been argued. Before these processes can be modeled, they need to be characterized and understood. [8].

Large-scale telecommunication outages can result in heavy losses to business and society at large. Also, colocation and the resulting concentration of telecommunications assets represent "serious risk posed to small and mid-sized businesses from disruptions in telecommunications service." [9]. Local switches are in end offices that can represent concentrations. Additionally, although end offices are susceptible to power loss, they are typically protected by generator and battery backup power sources. However, damages to the AC power grid are common in hurricane prone areas, which causes outages to "wire-line networks, wireless networks, transmission links, cable TV grids, and TV and radio facilities...". The major causes of telecommunication outages from hurricane Katrina was found to be power loss because of fuel supply disruptions, flooding and security in low-lying areas [10].

Recently, Lyons, et al empirically assessed the economic impact of telecommunication outages [11]. In that paper, the economic costs of telecommunication outages, for fixed line networks, is estimated for both a complete sector outage and local exchange outages. The costs estimates are based on actual business and residential demographics, including service and manufacturing areas. For seven local exchange outages in Ireland, these costs ranged from €370,000 to €1.1 million per day. Unfortunately, the number of lines for local switch in each of the exchanges are not given.

The User Lost Erlang (*ULE*) was introduced by McDonald in [12] as an impact metric for large-scale outages. The *ULE* is the logarithm of the magnitude of an outage, or $ULE = log_{10}(Magnitude)$, where magnitude is the number of users impacted. For instance, the impact of an outage affecting 10,000 lines would be *4 ULE*. Because of the range of outages could be many orders of magnitude, McDonald thought such a metric would be easy to use and understandable by the public, much like the logarithmic Richter scale for earthquake intensity. Of course, the disadvantage of the *ULE* is that although size is taken into account, the duration of the outage is not. So an outage has both size and duration as variables. The Federal Communication System used the Lost Line Hour (*LLH*) metric, the product of the number of lines out times the duration in hours. For instance, a 10,000 line switch out for 2 hours would be an equivalent outage of 20,000 *LLH*, or 20,000 lines out for an hour. Although the *LLH* incorporates both the time and duration of an outage, it does not capture blocked calls. Additionally, the *LLH* has no logarithmic transformation to tame outliers in size or duration of outages. Committee T1 in the US, published an American National Standards Institute (ANSI) sponsored metric called the Outage Index (*OI*). This metric included the product of two weightings, one for size, and another for duration. Additionally, given the long tailed distributions of size and duration, each weight included logarithmic transformations. However, further analysis of the *OI* indicated a network administrator perspective rather than a subscriber perspective, in that the index was insensitive to long duration outages and very sensitive to large size outages [13]. For instance, assume a local switch with 10,000 lines experiences an outage: if a 1 day outage the *OI* is 0.519 while if an 8 day outage the *OI* is 0.532 [14]. This discounting of duration masked significant impact of outages below a regulatory reporting threshold, which according to the *OI* impact metric was minimal [15].

More recently, in [16], resiliency was defined as the percentage of users deriving successful service over time. Although a reasonable metric, the number of users impacted is not provided by this metric.

B. Local Switch Reliability

In the late 90s, Kuhn reported on the sources of major outages in the PSTN, including local switches, tandem switches, signaling, and transmission facilities. However, this work assessed but a few years of outages, and only those outages that exceeded impacting 30,000 or more subscribers for at least 30 minutes or more [17].

Time series analysis on local switch outages lasting 2 minutes or more was first reported by Snow in [18], examining trends, causes and impact of outages occurring from 1992 to 1995. In [19], Snow noted that some individual switches seemed to experience many outages. Recently, local switch outages were examined by Snow, et al in [1], from 1996 to 2009, extending the work in [18] but

 ISBN: 978-1-61208-318-6

with the primary focus on switches suffering more frequent outages. Although summary statistics for causality and survivability were presented, causality trends, and the corresponding outage impact trends, were not investigated.

C. Reliability Assessments of Repairable Systems

A system is reliable when it carries out its intended functions over a specified period of time, without failure. The probability of this happening is the definition of reliability. Said another way, reliability is the probability a system will perform its intended function, in the intended environment and at a particular level of performance. Thresholds are very commonly used to declare a system as in either an "operational" or "degraded" mode [20]. Others define reliability as "conformance to specifications over time" [21].

As pointed out by Louit, et al. [22], reliability of systems are best assessed by time to failure (*ttf*) models and analysis. Because local switches are repairable systems, stochastic point processes must be applied, since the *ttfs* can be systemically changing over time. Said another way, the failure arrival process might be non-stationary, so *ttfs* cannot simply be fitted to a distribution unless they can be shown to be a renewal process (RP, independent and identically distributed) or a special case of renewal process, the homogeneous Poisson process (HHP, independent and identically exponentially distributed). So, the presence of trends are of paramount interest as they indicate improvement or deterioration.

Additionally, Louit, et al. [22], point out that methods for trend analysis are graphical and analytical. Graphical methods culminate in visual assessments of cumulative plots of outages versus time. Straight lines represent constancy, or no trend. Where the curve that bends up, represents an upward trend, and where a curve bends down, represents a downward trend of failures over time. Analytic trends are indicated when the visual trend evidence from the graph is slight, or to assess the strength of the trends. The Laplace trend test is well known, having a null hypothesis of HPP and an alternative hypothesis of a non-homogeneous Poisson process (NHPP). For no trend, renewal processes are often used, as it is hoped the system is made "good as new" through modular replacement. However, switches involve software and hardware changes, wherein some repairs result in a slightly different switch. This means that failures are not independent or identically distributed. The NHPP, is the most popular nonstationary model used in reliability for monotonically increasing or decreasing trends.

The Laplace statistic is zero for the null hypothesis of no trend. A statistic less than zero is a decreasing trend (reliability growth) while a statistic greater than 0 indicates an increasing trend (reliability deterioration). The Laplace score (L) asymptotically approaches a normal score, so if one chooses a critical value of 0.05, L is non-zero, representing a statistically significant trend, if $-1.96 \geq L \geq +1.96$. The Laplace score for a truncated time series of events is:

$$L = \frac{\sum_{i=1}^{n} t_i - \left(\frac{1}{2}\right)nT}{\sqrt{nT/12}} \qquad (1)$$

where t_i is the time of the i^{th} failure, n is the number of failures over the time observed time period T.

Lastly, Louit, et al. [22], point out that in cases as this study, where failures are from thousands of different local switches of different manufacturer and models, the superposition of many processes tend to converge to a Poisson process, either homogeneous or nonhomogeneous.

III. RESEARCH QUESTIONS

This research addresses the following questions regarding telecommunication local switch outage causality trends over a 14 year period:

- Are failure trends the same or different for different causal categories?
- Can causality failure processes be characterized as HPP or NHPP?
- Can a new impact metric be devised to provide insights into resiliency trends?
- For each causal category, are resiliency trends discernable from reliability and outage impact trends?

In this work, the PSTN is viewed as a single system, made up of switching, signaling and transmission segments. The switching segment is made up of the tandem and local switch subsystems. The purpose of this paper is to investigate the reliability and resiliency trends of the local exchange switching subsystem as a whole, by investigating the pooled failures of all individual local switches in the PSTN. There are a large number of different manufacturers and models of local switches in this infrastructure. Even the same model switch varies substantially from serial to serial because of differences in customers served and features offered. By pooling failures from different switches, we may assess the resiliency and reliability of local switching as a whole, rather than the reliability of a single switch. Then, failures and outages can be subdivided by causality and examined further for trends.

IV. TIME SERIES ANALYSIS METHODS

Two different time series methods are used. The first is the aforementioned graphical and analytical time-to-failure techniques, of Louit, et al. [22]. The second consists of methods developed for this paper: graphical plots of outage resiliency and linear regression, where indicated by visual assessment.

A. Causal Trend Analysis of Events

By combing similar cause codes reported to the FCC into categories, we reduced the fifteen causes reported to the FCC down to six causality codes:

- Scheduled outage: An intentional outage for maintenance purposes.

 ISBN: 978-1-61208-318-6

- Procedural error: Procedural errors made in installation, maintenance or other activities by Telco employees, contractors, switch vendors, or other vendors.
- Design error: Software or hardware design errors made by the switch vendor prior to installation.
- Hardware error: A random hardware failure, which causes the switch to fail.
- External circumstances: An event not directly associated with the switch, which causes it to fail or be isolated from the PSTN.
- Other/unknown: A failure for which the cause was not ascertained by the carrier.

These categories, their composition, and the distribution of failures to each category are shown in Table I. For each of these processes, time series were created for study over a 14 year period.

B. Causal Resiliency Trend Analysis

Manifold shortcomings of the aforementioned outage index indicate a different resiliency metric is needed for this analysis. Given the long tailed distribution of both switch lines and outage duration, *LLH* ranges from very small to very large, indicating a logarithmic transformation is desirable. However, a major problem with the outage index is a non-intuitive lack of reference and insensitivity to long duration outages, while a major advantage of *LLH* is that it represents switch size and duration equally.

TABLE I. LOCAL SWITCH OUTAGE CAUSE CATEGORY DISTRIBUTION

Outage Category	No.	%
Scheduled	3,885	30%
Procedural Error	1,394	11%
Design Error	1,214	9%
Random HW Failure	2,951	23%
Ext. Circumstances	2,900	23%
Other/Unknown	516	4%
Total	12,860	100%

To develop a new metric, we borrow from the field of communications engineering, where power is represented by decibels, referenced to a power level of interest. For instance, *dBm* is power in decibels referenced to one milliwatt (mW), while *dbW* is power represented to one watt. For example:

$$dBm = 10log_{10}\frac{P}{1\,mw} \qquad (2)$$

The nice feature about the *dBm* is that 0 *dBm* is 1 mW (the reference power), and a *10 dB* increase/decrease is a tenfold increase/decrease, and a *3 dB* increase/decrease represents a doubling /halving.

The new metric used here is as follows:

$$OI_{dBK} = 10log_{10}\frac{LLH}{1000} \qquad (3)$$

This new metric is called an outage index, referenced to *1,000 LLH*. So now, an OI_{dBk} of *20* represents two orders of magnitude above *1,000 LLH*, or *100,000 LLH*, while an OI_{dBK} of *23* represents a doubling above *20*, or *200,000 LLH*. Also, -3 OI_{dBK} represents *500 LLH* while -6 OI_{dBK} represents *250 LLH*. This new metric should "tame" the wide swings of *LLH*, and give an intuitive reference when doing time series plots and regression of outage resilience over time. Of course, if desirable, we can also have OI_{dBM} which references the severity to one million *LLH*.

V. RESULTS

Here we present tabular, graphical, and analytic results to assess reliability and resiliency and causal trends. First, the cumulative outage plot for all outages is seen in Figure 1. This is not a monotonic trend, as the failure rate (failures per unit time, the derivative of the cumulative graph) represents a "bathtub" curve: a region of high failure rate, followed by a region of lower failure rate, then an increasing failure trend.

A. Causal Reliability Trends

A summary of causal trend analysis is provided in Table II, while sample casual trend graphic results are shown in Figures 2 through 4 show sample cumulative plots. In Table II, we see three cause categories that show overall improvement and three that overall show deterioration.

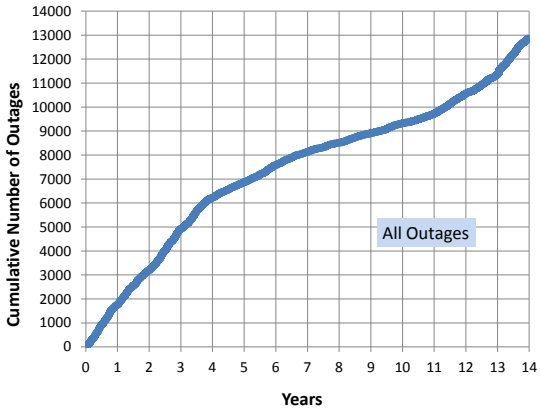


Figure 1. Cumulative Outage Plot: All Outages

TABLE II. OUTAGE FREQUENCY TRENDS BY CAUSE

Cause Category	L	p-Value	Trend	Type
Scheduled	-58.41	0.0000	Decreasing	Monotonic
Procedural Errors	-14.42	0.0000	Decreasing	Monotonic
Design Errors	-27.61	0.0000	Decreasing	Monotonic
Random HW Fail.	3.0	0.0012	Increasing	Bathtub
External Circum.	28.33	0.0000	Increasing	Monotonic
Unknown/Other	3.34	0.0004	Increasing	Bathtub

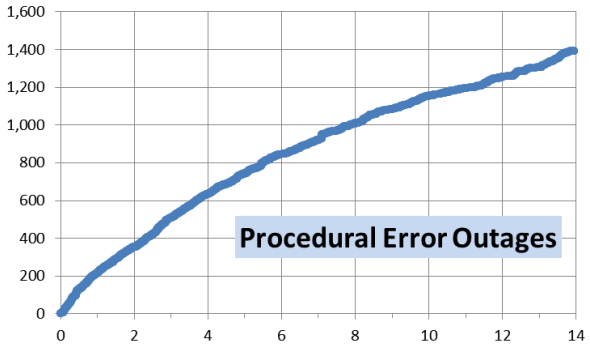


Figure 2. Cumulative Outage Plot: Procedural Outages

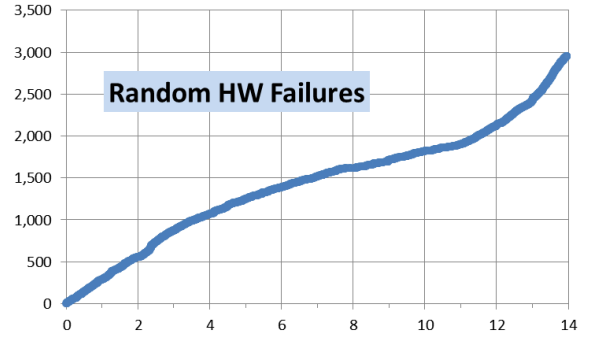


Figure 3. Cumulative Outage Plot: Random HW Failures

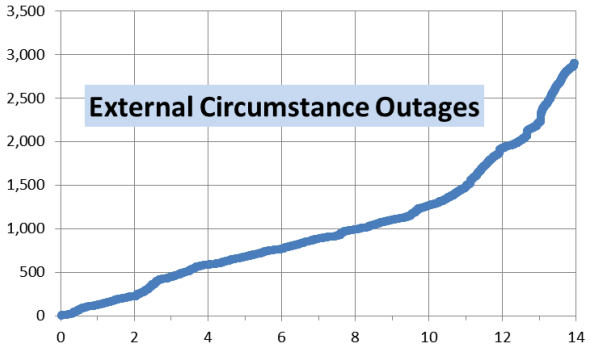


Figure 4. Cumulative Outage Plot: External Circumstance Outages

These trends are statistically very strong, as evidenced by the p-values. Several categories showed monotonic trends, good candidates for NHPP. Two showed bathtub type failure rates, and could be examined in a piecewise linear fashion. Interestingly, none showed promise of a renewal process, for which distributions could be fitted, as all six processes are nonstationary over the 14-year study period.

B. Causal Impact Trends

A summary of causal impact trends is shown in Table III, where we observe instances of increasing, constant and decreasing outage impact. The trends were determined by linear regression models, which were all statistically significant at the *0.05* critical value, for each model's F-test and coefficient t-tests. The trend constant and slope are provided for each model. Several observations can be made from this table. First, note that since the scheduled outage constant is about *3 dBK* higher than design error constant, on average, scheduled outage severity is about double that of design error. Secondly, however, using the regression constant, note that over 14-years, scheduled outage resiliency improved *17 dBK* while design error resiliency improved *13.2 dBK*. So by the end of the period, the aforementioned *3 dBK* difference evaporated. Lastly, note that over the 14-year period, external circumstance resiliency worsened by *7.4 dBK*, meaning its worsening more than quadrupled (*6 dB*).

TABLE III. OUTAGE IMPACT TRENDS BY CAUSE

Cause Category	Impact Trend	Regression Const.	OI_{dbk}/Yr	OI_{dbk}/14Yr
Scheduled	Decreasing	26.3 OI_{dbk}	–1.22 OI_{dbk}	OI_{dbk}
Procedural Errors	No	NA	NA	NA
Design Errors	Decreasing	23.4 OI_{dbk}	–0.94 OI_{dbk}	OI_{dbk}
Random HW Fail.	No	NA	NA	NA
External Circum.	Increasing	24.9 OI_{dbk}	0.53 OI_{dbk}	OI_{dbk}
Unknown/Other	No	NA	NA	NA

Next, we show the impact charts (OI_{dBK} quarterly plots), Figures 5 through 8, corresponding to the four cumulative outage plots, followed by an example *LLH* plot for outages due to external circumstances (Figure 9). The *LLH* plot in Figure 9 not only demonstrates the difficulty in determining trends, but also that *LLH* is a poor resiliency impact metric, compared to Figure 8.

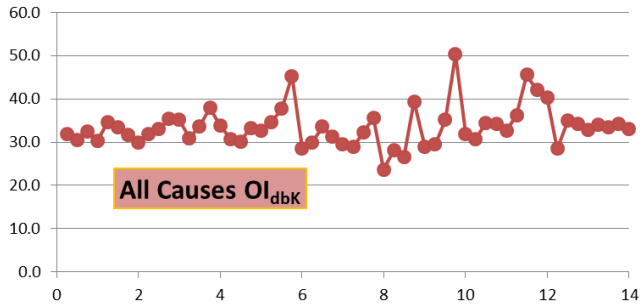


Figure 5. Outage Impact Plot: All Outages

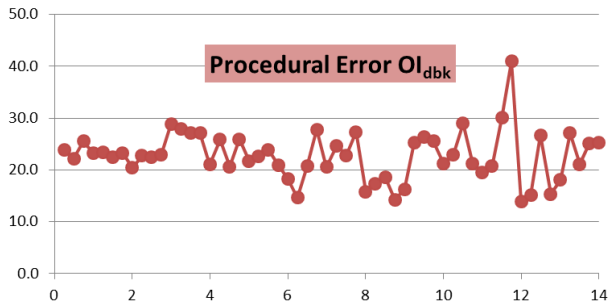


Figure 6. Outage Impact Plot for Procedural Error Outages

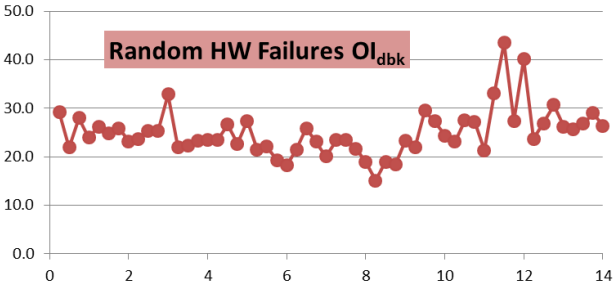


Figure 7. Outage Impact Plot: Random HW Failures

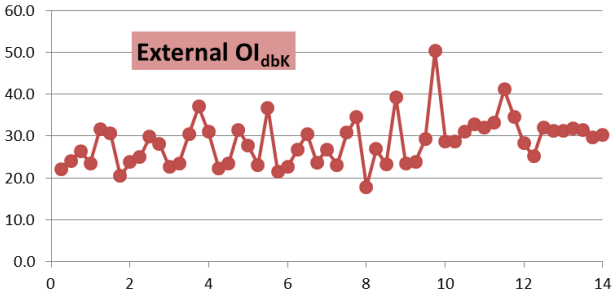


Figure 8. Outage Impact Plot: External Circumstance Outages

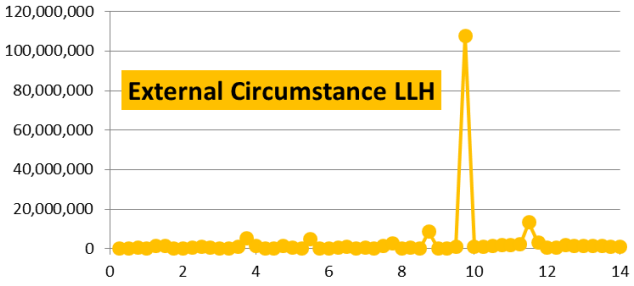


Figure 9. LLH Plot for External Circumstance Outages

VI. SUMMARY OF FINDINGS

The research questions listed in Section III are addressed here in turn.

A. *Are outage/failure trends the same or different for different causal categories?*

All failure/outage trends were markedly different. However, three causality trends were found to be decreasing over the study period, indicating reliability growth. Likewise, three other causality trends were found to be increasing, indicating reliability deterioration. In addition, some of the trends were monotonically increasing or decreasing, and others indicated bathtub processes, which indicate deterioration towards the end of the study period.

B. *Can causality failure processes be characterized as either HPP or NHPP?*

None of the six processes studied passed the test for HPP as all showed strong visual signs of nonstationary processes. The monotonically increasing and decreasing trends looked like good candidates for the power law model. The causality trends exhibiting bathtub characteristics are candidates for investigating piecewise linearity, or piecewise HPP.

C. *Can a new impact metric be devised to provide insights into resiliency trends?*

The new impact metric, the OI_{dBK}, showed promise as it controlled outliers through a logarithmic transformation and easily allowed trend analysis for resiliency. Also, it is referenced to a benchmark loss of *1,000 LLH*, making it intuitive. This metric overcomes shortcomings in both the *LLH* and the ANSI outage index metrics. Unlike the ANSI outage index, it is not insensitive to long duration outages, giving equal weight to both outage magnitude and duration. Unlike the *LLH*, it does not have extreme range of values.

D. *For each causal category, are failure/outage trends and impact trends in agreement?*

For scheduled and design error outage categories, both impact and outage trends decreased (deterioration in both) while for external circumstance outages, both trends increased (improvement in both). Interestingly, although procedural error outages decreased, there was no improvement in outage impact. In addition, random hardware failures and unknown/other outages increased (deterioration), with no concomitant deterioration outage impact.

VII. CONCLUSIONS

This work shows that to assess resiliency, both reliability and outage impact offers valuable insights. It also indicates the importance of posterior perspectives of these dependability attributes, as an important part of the fault management aspect of the network management function.

More research can be performed to gain additional perspectives. For instance, besides investigating the trends

 ISBN: 978-1-61208-318-6

for the casual categories here, all fifteen cause codes can be examined for resiliency trends. Preliminary investigations indicates presence of possible renewal processes, including HPP. For those cases, event independence can be investigated using the coefficient of variation and presence of significant first order autocorrelation coefficients. Additionally, piecewise linear and power law modeling could be used to further characterize these causal processes.

REFERENCES

[1] Andrew Snow, Aimee Shyirambere, Julio Arauz, and Gary Weckman, "A reliability and survivability analysis of local telecommunication switches suffering frequent outages", The Twelfth International Conference on Networks, IARIA, Seville Spain, 2013.

[2] J. Gillan, and D. Malfara, The transition to an all-IP network: a primer on the architectural components of IP interconnection, National Regulatory Research Institute, May 2012.

[3] FCC, Local telephone competition: status as of december 31, 2010, Industry Analysis and Technology Division, Wireline Competition Burea, October 2011.

[4] P. D. T O'Conner, Practical reliability engineering, Fourth edition, John Wiley & Sons, England, 2001.

[5] R. W. A. Barnard, "Reliability engineering : futility and error", Second Annual Chapter Conference, South African Chapter, International Council on Systems Engineering (INCOSE), 31 August, September 2004.

[6] FCC Report 43-05, ARMIS Service Quality Report Table IVa, downloaded from http://transition.fcc.gov/wcb/armis/ September 2012.

[7] B. Allenby and J. Fink, "Toward inherently secure and resilient societies," Science 309, August 2005, pp. 1034.

[8] Laura J. Steinberg, Nicholas Santella, and Corri Zoli, "Rouge post-katrina: the role of critical infrastructure modeling in promoting resilience", Homeland Security Affairs, Vol 7, Article 7, 2011.

[9] Ginger Armbrustera, Barbara Endicott-Popovsky, and Jan Whittington, "Are we prepared for the economic risk resulting from telecom hotel disruptions?", International Journal of Critical Infrastructure Protection Volume 5, Issue 2, July 2012, pp 55–65.

[10] A. Kwasinski, W. W. Weaver, P. L. Chapman, and P. T. Krein, "Telecommunications Power Plant Damage Assessment Caused by Hurricane Katrina – Site Survey and Follow-Up Results", 28th Annual International Telecommunications Energy Conference, INTELEC '06, IEEE, ISBN: 1-4244-0430-4, 2006.

[11] S. Lyons, Edgar Morgenroth, and Richard Tol, "Estimating the value of lost telecoms connectivity", Electronic Commerce Research and Applications 12, 2013, pp. 40–51.

[12] J. C. McDonald, "Public network integrity-avoiding a crisis in trust*"* Journal on Selected Areas in Communications, IEEE Journal, Volume: 12 , Issue: 1, 1994.

[13] A. Snow, "A survivability metric for telecommunications: insights and shortcomings", IEEE Computer Society, Proceedings, Information Survivability Workshop – ISW'98, 1998, pp. 135-138.

[14] A. Snow and Y. Carver, "Carrier-industry, fcc and user perspectives of a long duration outage: challenges in characterizing impact", T1A1.2/99- 026, Contribution to Committee T1 – Telecommunications, Boulder Colorado, 1999.

[15] A. P. Snow, "Assessing pain below a regulatory outage reporting threshold", Telecommunications Policy, Vol. 28, Issue 7-8, 2004, pp. 523-536.

[16] M. Omer, R. Nilchiani, and A. Mostashari, "Measuring the resilience of the global internet infrastructure system", 3rd Annual IEEE Systems Conference, March 2009, pp. 152-162.

[17] R. Kuhn, Sources of failure in the public switched telephone network, IEEE Computer, 1997.

[18] A. P. Snow, "The reliability of telecommunication switches", Six International Conference on Telecommunications Systems: Modeling and Analysis, March 1997, pp. 288-295.

[19] A. P. Snow, "Internet implications of telephone access", IEEE Computer 32 (9) , September 1999, pp.108-110.

[20] M. A. Levin and T. T. Kalal, Improving Product Reliability, Strategies and Implementation, John Wiley & Sons, England, 2011.

[21] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. Longstaff, and N. R. Mead, Survivable Network Systems: An Emerging Discipline, Carnegie-Mellon Software Engineering Institute Technical Report CMU/SEI-97-TR-013, 1997, revised 1999.

[22] D. M. Louit, et al, "A Practical procedure for the selection of time-to-failure models based upon the assessment of trends in maintenance data", Reliability Engineering and System Safety 94, 2011, pp. 1618-1628.

 ISBN: 978-1-61208-318-6

DDoS Attack Detection Using Flow Entropy and Packet Sampling on Huge Networks

Jae-Hyun Jun
School of Computer Science and Engineering
Kyungpook National University
Daegu, Republic of Korea
jhjun@mmlab.knu.ac.kr

Dongjoon Lee
School of Computer Science and Engineering
Kyungpook National University
Daegu, Republic of Korea
djlee@mmlab.knu.ac.kr

Cheol-Woong Ahn
Division of Digital Contents
Keimyung College University
Daegu, Republic of Korea
homepig@kmcu.ac.kr

Sung-Ho Kim
School of Computer Science and Engineering
Kyungpook National University
Daegu, Republic of Korea
shkim@knu.ac.kr

***Abstract*— While the increasing number of services available through computer networks is a source of great convenience for users, it raises several concerns, including the threat of hacking and the invasion of user privacy. Hackers can easily block network services by flooding traffic to servers or by breaking through network security, hence causing significant economic loss. It is well know that a Distributed Denial of Service (DDoS) attack, which robs the targeted server of valuable computational resources, is hard to defend against. In order to address and nullify the threat to computer networks from DDoS attacks, an effective detection method is required. Hence, huge networks need an intrusion detection system for real-time detection. In this paper, we propose the flow entropy- and packet sampling-based detection mechanism against DDoS attacks in order to guarantee normal network traffic and prevent DDoS attacks. Our approach is proved to be efficient via OPNET simulation results.**

Keywords-packet sampling; flow entropy; ddos detection; Network Security;

I. INTRODUCTION

Novel and ever-varying network services are being developed and launched as the rapid growth of the Internet and online users continues. A 2009 investigation by the German company Ipoque shows that Peer-to-Peer (P2P) traffic has constituted more than 60% of Internet traffic for the last few years, and will be responsible for a sizeable portion of it in the foreseeable future [1].

While the Internet provides numerous services through computer networks that make our lives easier, this convenience comes at the cost of ever-rising Internet crime, generally, in the form of hacking and similar invasions of privacy. These crimes cause significant economic damage by flooding network servers or hindering services by gaining access to the relevant computer systems [2].

The Distributed Denial of Service (DDoS) is not a new attack technology. While it first appeared in the late 1990s, the first well-publicized DDoS attack occurred in 2000 against major Internet corporations including Yahoo, Amazon, CNN and eBay. It has been more than ten years since a major DDoS attack has occurred. However, DDoS attacks are among the greatest threats for Internet infrastructure and for the information technology environment.

A DDoS attack occurs when the intruder, also called the *attacker*, invades one or more systems online. The initially compromised system is typically one with a large number of users and a high Internet bandwidth. The attacker then installs the attack programs on the initially compromised system, called the *DDoS master*. The master is then used to find other systems on the network that are vulnerable, and installs DDoS agents, called *daemons*, on these. Using the master system, the attacker then instructs the DDoS daemons to attack the intended target, or *victim*, of the DDoS attack. Hence, the conceptual node of a DDoS attack is comprised of attacker, master, daemon or zombie, and victim. Table 1 shows the explanation of each node. The structure of a DDoS attack is represented in Figure 1 [3].

Since it is not easy to distinguish between a DDoS attack and normal traffic, it is possible to misjudge a normal data packet as a DDoS attack packet. Thus, in order to protect a system from DDoS attacks, a method for an accurate analysis of incoming traffic and the detection of a DDoS attack therein takes pragmatic precedence.

TABLE I. ROLE OF DDOS ATTACK NODES [3]

NAME	ROLE
Attacker	Attacker who is leading all attack operates with an instrument by remote control and delivers commands directly.
Master	Master receives the commands from attacker and orders attack zombies managed by this master.

 ISBN: 978-1-61208-318-6

Zombie	They are controlled by master. Attack program operates the commands that came from each master, and finally performs their attack to the victims.
Victim	As for final victims, simultaneously they are attacked from several hosts.

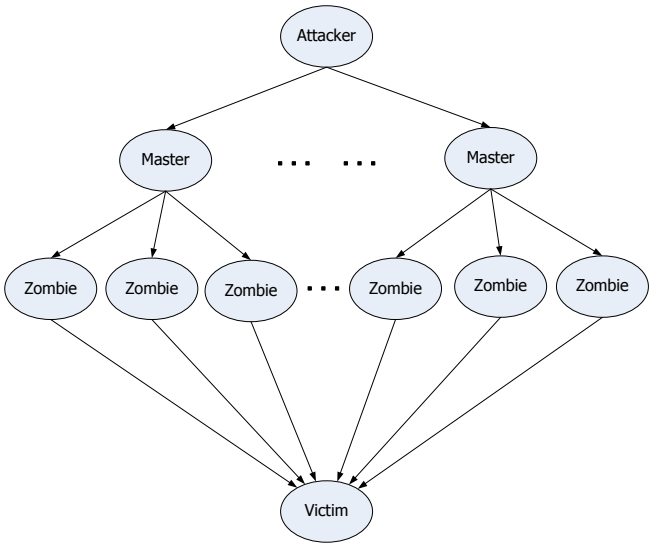


Figure 1. The structure of DDoS attack

Figure 2 shows the annual number of DDoS attacks on Korea Internet & Security Agency (KISA) [11]. As shown, in 2010, there were 6 small-scale 1Gbps DDoS attacks, 4 middle-scale attacks each within the 1~5Gbps and 5~10Gbps bandwidth ranges respectively, and 10 large-scale attacks of more than 10Gbps. By contrast, in 2012, 56 small-scale DDoS attacks occurred within the 1Gbps range, 21 and 25 middle-scale attacks within 1~5Gbps and 5~10Gbps respectively, and 36 large-scale attacks of bandwidth over 10Gbps. It is evident, then, that DDoS attacks are increasing in number by the year.

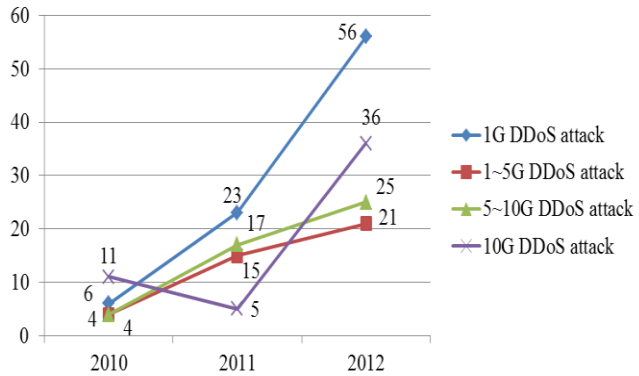


Figure 2. Number of annual DDoS attack on KISA [11]

This paper is structured as follows. In Section 2, we introduce DDoS attack and detection methods. In Section 3, we explain the DDoS attack detection method that uses using flow entropy and packet sampling on a large network, and the results of the testing and evaluation of this method are presented in Section 4. In the final section, we will reflect on our findings.

II. RELATED WORK

A. DDoS background

DDoS attacks first emerged as a kind of massive traffic-generation attack. In some of the first ones of this sort, attackers were able to harness a large amount of network traffic and transmit them to the target systems. In 2000, Yahoo and Amazon's web sites were targeted with such DDoS attacks. Several tools, such as Tribe Flood Network (TFN), TFN 2000 (TFN2K), Trinoo, Stacheldracht, etc., were employed for this type of attack. At the time, researchers were developing traffic anomaly detection techniques for huge networks with a lot of traffic. While it was possible to detect DDoS-type attacks with network traffic anomaly detection techniques on account of higher-than-usual bandwidth usage, it was very difficult to effectively block them. This was because even if a DDoS attack was detected, there was no way to accurately identify the specific attack packets due to IP address spoofing techniques.

Internet worms attack vulnerable systems and take them over automatically. In one such attack, the now-infamous 'Slammer Worm' infected more than 75,000 machines in 10 minutes, causing several network servers worldwide to crash. Figure 3 depicts the global scale of the outbreak [4].

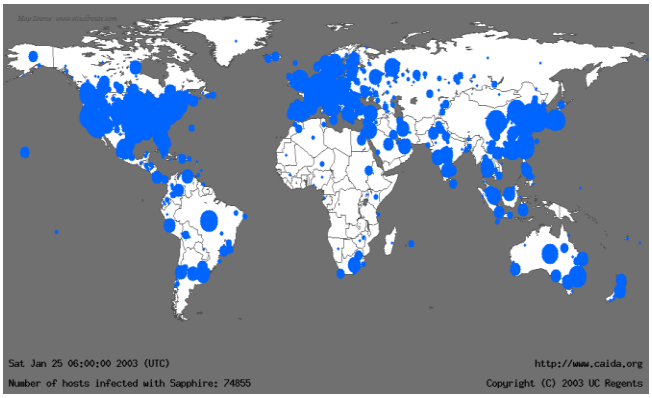


Figure 3. Geographic propagation of Slammer worm 30 minutes after release [4].

Since the mid-2000s, DDoS attack trends have changed. These days, DDoS attacks are primarily launched for economic gain. For instance, a hacker may demand payment from a company in exchange for not attacking its systems. For attacks of this sort, the hacker would prefer not to paralyze entire networks, but would only want the application-layer service to be unavailable to users. In order to do this, the attacker does not have to generate a massive amount of traffic. In fact, if the attack traffic is generated in a sophisticated manner, application-layer services could easily be brought down using only several kbps of attack traffic bandwidth. The attack data packet in this case resembles a

 ISBN: 978-1-61208-318-6

normal packet. It is thus, very difficult to detect an application-layer DDoS attack using only attack-traffic bandwidth analysis or packet-based attack detection methods, which, nowadays, are widely used to defend against DDoS attacks.

A pertinent instance of the above issue is a large DDoS attack that lasted from July 5 to July 10, 2009, launched against 48 websites in the United States and South Korea using tens of thousands of zombie PCs. The attackers in this case used numerous techniques, such as TCP SYN flooding, UDP/ICMP flooding, HTTP GET flooding, and CC attacks. While these attacks were being attempted, the ones on the HTTP service were not effectively preventable. This is because almost all DDoS detection techniques are based on bandwidth variation and the volume of traffic. Even though the aggregate volume of attack traffic was huge, these techniques failed to detect the exact attackers because the amount of traffic from each attack system was not sufficiently high for it to be located. For application-layer DDoS attack detection [5], it is necessary to develop an application-behavior-based attack detection technique [6].

B. DDoS Detection research

Machine learning can be roughly divided into two parts: Supervised and Unsupervised learning methods. Supervised methods use labeled data for training. A supervised learning approach uses labeled 'training data' to classify traffic as normal or otherwise [9]. Unsupervised learning methods use unlabeled data samples. A typical example is clustering. When the data flows in, it clusters the data into different groups [7]. With the incoming data so divided, the program can then inspect and detect abnormal data packets, such as those used for DDoS attacks, by any of a variety of detection methods.

In [8], the flow is formed using a quintuple, which consists of source/destination IP addresses, source/destination port numbers and the protocol. The entropy of four of the features -- source/destination IP address and port number -- is calculated to form clusters. The information is saved to the entropy cube, based on the destination IP address. If the entropy values of the source IP address and port number are higher than a certain preassigned value, or if the entropy value of the destination port number is lower, the entropy cube labels them as a DDoS attack.

A classification problem arises if the entropy of heavy traffic has a value similar to that of a DDoS attack. Hence, we propose that incoming traffic be classified by using flow entropy as well as packet sampling of data.

III. PROPOSED METHOD

In order to detect DDoS attack flows on huge networks, we classify flow using packet sampling, as well as considering measures of flow entropy, the average entropy, the entropy of the source port and the number of packets/second. The flowchart in Figure 4 depicts our proposed method.

From incoming traffic, we extract one of every five data packets for sampling. Figure 5 shows the packet sampling on a router. The sampled packets are collected during a 'time window'.

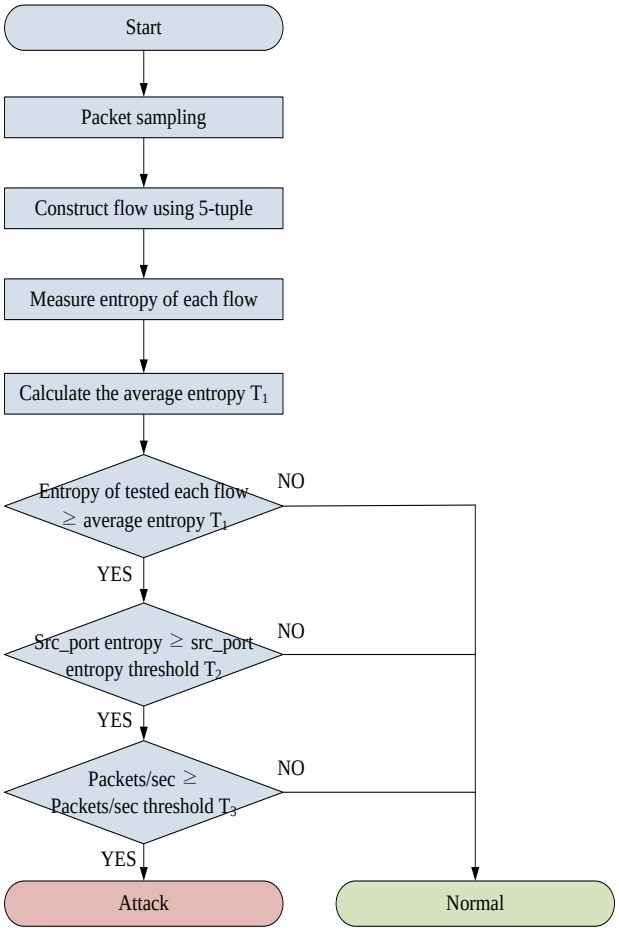


Figure 4. Proposed method

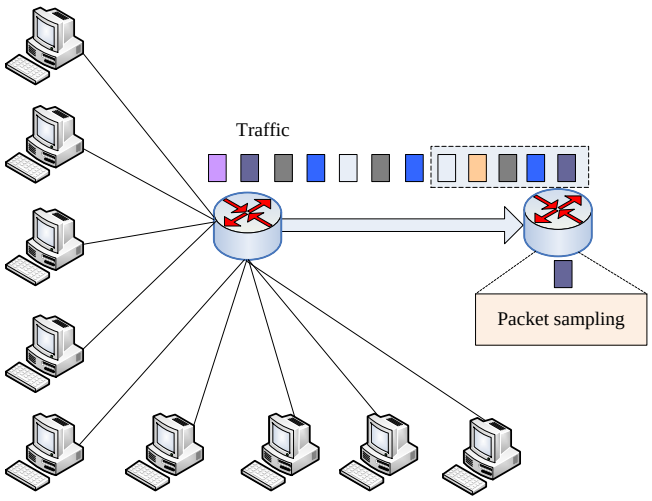


Figure 5. Packet sampling on router

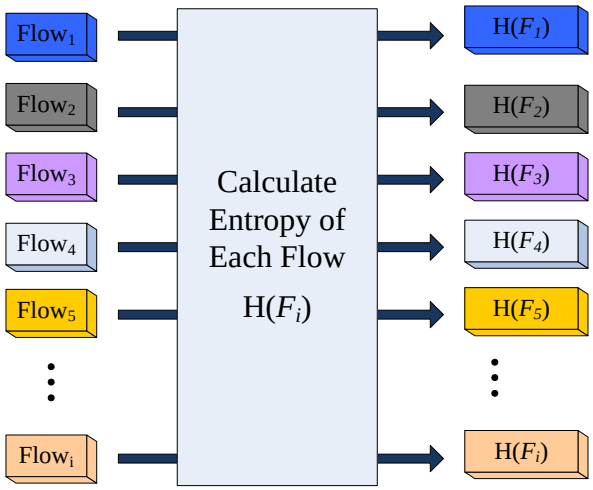


Figure 6. Measure entropy of each flow

The physical definition of entropy was offered by the German physicist Rudolf Clausius, in 1865 [10]. Since entropy causes uncertainty, it is impossible to accurately predict what happens next in an entropic situation. However, if entropy decreases, uncertainty decreases as well. In such cases, entropy may be calculated using the following equations:

$$H(F_i) = -\sum_{i=1}^{n} P_i \log_2 P_i \quad (1)$$

In Equation (1), n is the number of features (packet number, source port, packets/sec), P_i is the probability of feature i.

$$H(F_{avg}) = \frac{-\sum_{i=1}^{n} H(F_i)}{N(H(F_i))} \quad (2)$$

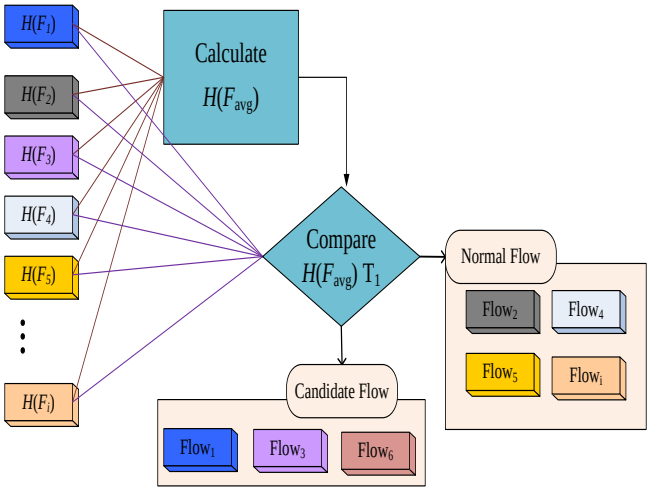


Figure 7. Calculate the average entropy T_1

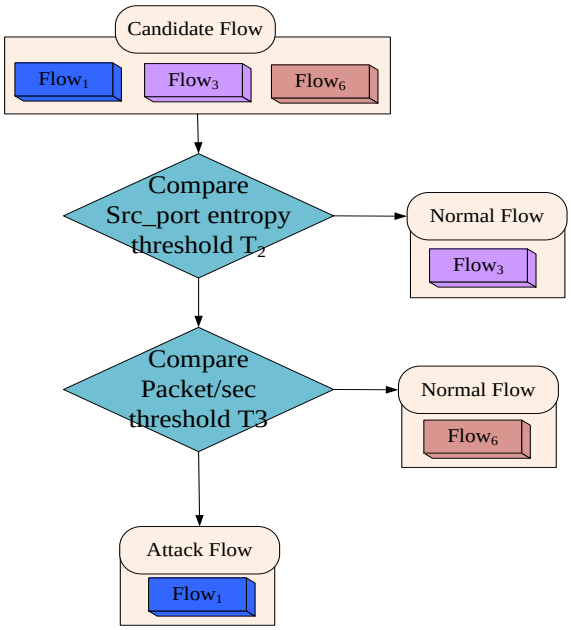


Figure 8. Compare source port entropy T_2 and packet/sec T_3 threshold

In Equation (2), $N(H(F_i))$ is the total flow. In general, DDoS attacks consist of several attack packets. In order to evaluate a flow with several sampled packets, we compare its entropy obtained from (1) with the average calculated entropy $H(F_{avg})$ of the system:

$$\begin{cases} H(F_i) \le H(F_{avg})\ T_1, & Normal\ Flow\ N(F_i) \\ H(F_i) > H(F_{avg})\ T_1, & Candidate\ Flow\ C(F_i) \end{cases} \quad (3)$$

Figure 7 shows the average entropy. If the entropy value of the flow $H(F_i)$ is larger than $H(F_{avg})$, we select it as a candidate flow, $C(F_i)$, for a DDoS attack. If $H(F_i)$ is less than or equal to $H(F_{avg})$, we classify it as a normal flow $N(F_i)$. As is evident, candidate flows have a higher probability of being DDoS attacks.

$$\begin{cases} Source\ port\ entropy\ of\ C(F_i) > Source\ port \\ entropy\ Threshold\ T_2, & Candidate\ Flow\ C(F_i) \\ Source\ port\ entropy\ of\ C(F_i) \le Source\ port \\ entropy\ Threshld\ T_2, & Normal\ Flow\ N(F_i) \end{cases} \quad (4)$$

The candidate flow $C(F_i)$ is used to calculate the entropy of the source port number. This entropy is then compared with the source port entropy threshold T_2. A higher entropy value means that a lot of ports are being used for transmission. A DDoS attack always involves the use of several ports to transmit a large number of packets. If the measured port entropy is higher than the entropy threshold (T_2), the corresponding traffic will be designated as a candidate flow $C(F_i)$ (see Equation (4) and Figure 8).

 ISBN: 978-1-61208-318-6

$$\begin{cases} Packet/second\ of\ C(F_i) > Packet/second\ threshold\ T_3,\ Attack\ Flow\ A(F_i) \\ Packet/second\ of\ C(F_i) \le Packet/second\ threshlod\ T_3,\ Normal\ Flow\ N(F_i) \end{cases} \quad (5)$$

In the final stage of the detection process, we calculate the rate of packet transmission (packets/sec) and compare it with the packet transmission threshold (T_3) to determine whether or not the corresponding traffic is part of a DDoS attack. If the packet transmission rate is higher than T_3, the flow in question will be classified as a DDoS attack. This process is consistent with Equation (5) and Figure 8.

IV. THE RESULT OF EXPERIMENT

In this section, we will evaluate the performance of our DDoS attack detection method. Our method is applied to a 'victim' router. We use OPNET [12] to simulate the network environment and evaluate our approach.

A. Experiment circumstance

We allowed web services and e-mail traffic as normal traffic on the network. In addition, we used attack traffic to simulate DDoS attacks. The topology of the experiment is a star, and uses 50 nodes, 1 server and 3 routers.

We allocate the nodes as follows: there are 25 nodes (node 1~25) which create the DDoS attack traffic and send it to the server, while 22 nodes (node 26~47) constitute normal traffic; three nodes (node 48~50) act as the server. We collect the traffic in the router using a 6-second time window. We also set appropriate thresholds for average entropy (T_1), the entropy of the source port (T_2) and packet transmission (T_3).

B. Experiment result and analysis

In this section, we use OPNET to evaluate the performance of the proposed method.

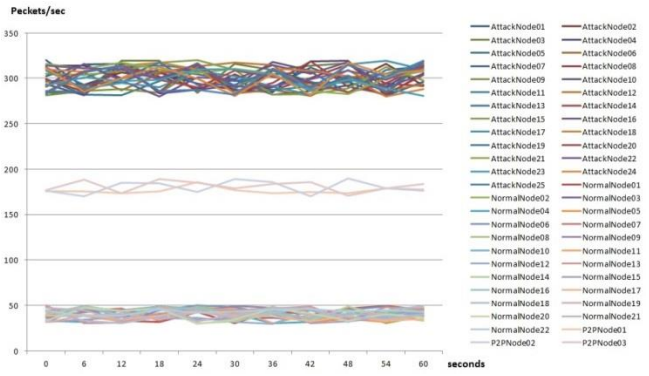


Figure 9. Creation rate each node packet

Figure 9 shows the relationship between packets transmitted (*y*-axis) and time (*x*-axis) for each node. Nodes 1 to 25, which simulated a DDoS attack, transmit approximately 300 packets per second, while nodes 26 to 47, used to imitate normal traffic, create around 40 packets per second. Nodes 48 to 50 -- our simulated P2P service -- create approximately 180 packets per second.

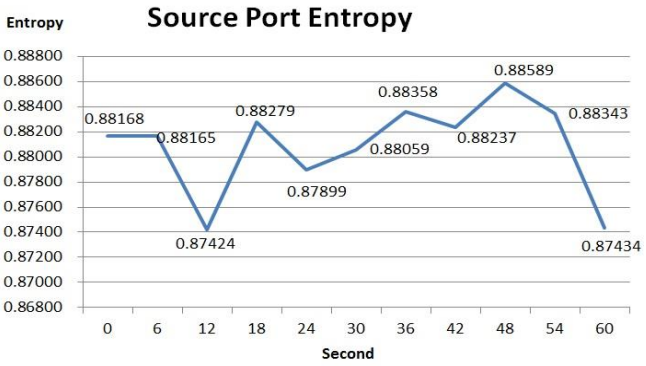


Figure 10. The entropy of source port number on candidate traffic

Figure 10 shows the entropy of the source port, which is used to determine whether traffic in question is a candidate for a DDoS attack. As we can see in Figure 10, the entropy of the source port is approximately 0.88, which is higher than the threshold value (T_2=0.8), as shown in Figure 10. Thus, it can thus be evaluated as candidate flow.

The last process involves checking the packets transmission rate of the candidate flow. As we can see in Figure 9, the rate for attack nodes is around 300 packets per second, far higher than the threshold (T_3=60). We can, thus, conclude that the flow is a DDoS attack.

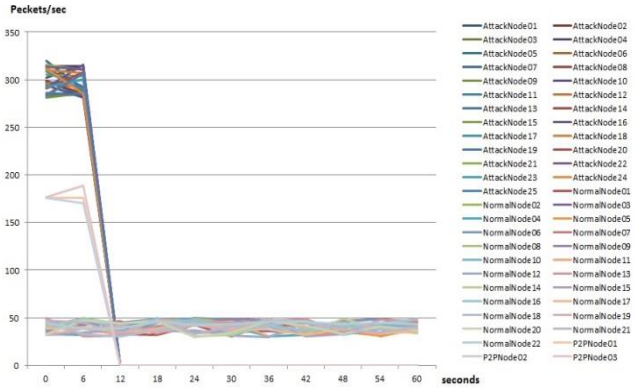


Figure 11. The previous method result

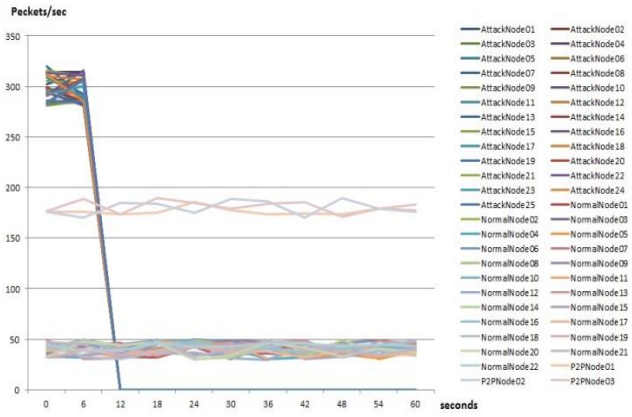


Figure 12. The Proposed method result

Figures 11 and 12 show the comparison between an existing DDoS attack detection method [8] and our proposed method. We can see that, unlike the existing ones, our proposed method can accurately detect DDoS attacks even in environments constituting small volumes of network traffic.

V. CONCLUSION

People gain much convenience from computer network because of the increasing services based on Internet. However, it is a "Double–edged Sword". It brings negative influence, such as Internet crime simultaneously. Every year, flooding attack causes a lot of economical loss.

In this paper, we proposed an effective DDoS attack detection method using flow entropy and packet sampling on a huge network. Once the DDoS attack is detected, we are able to control the attacker hosts. We have also demonstrated the superiority of our method to existing DDoS detection algorithms through experimental results.

REFERENCES

[1] G. Szabo, I. Szabo, and D. Orincsay, "Accurate Traffic Classification," IEEE Int. Symposium on World of Wireless Mobile and Multimedia Networks, 2007, pp. 1-8.

[2] Y. Xie and S. Z. Yu, "Monitoring the Application -Layer DDoS Attacks for Popular websites," IEEE/ACM Trans on Networking, vol. 17, no. 1, Feb. 2009, pp. 15-25.

[3] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems," ACM Computing Surveys, vol. 39, Iss. 1, Article 3, April 2007.

[4] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "The Spread of the Sapphire/Slammer Worm," http://www.caida.org/publications/papers/2003/sapphire/sapphire.html [retrieved; Dec 2013]

[5] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly, "DDoS-Shield: DDoS Resilient Scheduling to Counter Application Layer Atttacks," IEEE/ACM Transactions on Networking, vol. 7, no. 1, Feb. 2009, pp. 26-39.

[6] Arbor Networks ASERT Team, "July, 2009 South Korea and US DDoS Attacks," ARBOR Networks, July 2009.

[7] Y. Hong, S. Kwong, Y. Chang, and Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," ScienceDirect, vol. 41, no. 9, Sep. 2008, pp. 2742-2756.

[8] X. Kuai, Z. Zhi, and B. Suratol, "Profiling Internet Backbone Traffic Behavior Models and Application," ACM SIGCOMM, vol. 35, no. 4, Oct. 2005, pp. 169-180.

[9] T. Thapngam, S. Yu, and W. Zhou. " DDoS Discrimination by Linear Discriminant Analysis (LDA)," Computing, Networkng and Communications (ICNC) 2012, May 2012, pp. 532-536.

[10] R. Clausius, "The Mechanical Theory of Heat: With Its Applications to the Steam-engine and to the Physical Properties of Bodies," May 1867.

[11] Y. K. Park, "DDoS Attack Trend Analysis of 2012 year through the Cyber-Shelter," Internet and Security Focus, vol. 2, 2013.

[12] OPNET application and network performance, "http://www.opnet.com/"

 ISBN: 978-1-61208-318-6

Decision-Theoretic Planning for Cloud Computing

Rafael Mendes, Rafael Weingartner, Guilherme Geronimo, Gabriel Bräscher
Alexandre Flores, Carlos Westphall, Carla Westphall
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Brazil 88040-970
Email: {rafaeldesouzamendes,rafaelweingartner,guilherme.geronimo,gabrascher,alexandre.augusto.flores}@gmail.com,
{westphal,carlamw}@inf.ufsc.br

***Abstract*—This paper presents a mathematical model of decision planning for autonomic Cloud Computing based on the decision-theoretic planning model. It uses Markov decision process on the cloud manager to evaluate decisions and manage the Cloud environment. Also, it contributes to the state-of-art of Cloud Computing approaching the planning phase of the autonomic process with a mathematical model, considering two important factors, (1) the uncertainty of action's results and (2) the utility of the actions. Both factors are needed when dealing with complex systems as a Cloud.**

Keywords-cloud computing; decision-theoretic planning; autonomic computing; self-management

I. INTRODUCTION

The decision-theoretic planning (DTP) problems were extensively researched during the last decades. The main problem with the decision-theoretic (or probabilistic) approach for the planning phase in autonomic computing is the need to provide extensive information about the transitions between system states. However, with the arise of Cloud Computing (CC), sensor networks and other technologies that enabled the monitoring and collection of large volumes of data, the information became abundant and the recomendation of utility to solve the contradictions between rules on large rule-based polices [1], [2] must be taken seriously. On big data environments, the DTP problems no longer exist, enabling its application for the planning phase on the autonomic loop.

This work presents a model that plans actions for CC management systems using a decision-theoretic approach. It contributes to the state-of-art in CC research by:

- (i) Adapting the decision-theoretic models, which was based on *Markov Decision Process* (MDP), to use in the planning phase of the autonomic management loop;
- (ii) Introducing decision-theoretic and MDP for planning in CC;
- (iii) Applying mathematical models on a concrete decision making scenario for self-configuration of CloudStack [3].

This paper is organized as follows. Section II presents an overview of the concepts that are required to understand the proposed model. Section III presents the related works. In section IV is presented a conceptual proposal to guide the decision mechanism to CC.The Sections V and VI discusses and presents a mathematical model for the MDP approach, presenting a study case scenario of a Cloud implementation with CloudStack and Xen Cloud Platform. Finally, Section VII concludes the paper and presents future works that will improve the presented model.

II. LITERATURE REVIEW

A. Cloud Computing

After some years, the definition of CC that has grown in acceptance was created by NIST[4]:

> Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Another important contribution to CC also can be found in [5, Section 3]; "Cloud Computing: the need for monitoring", where are stated some useful concepts to understand the fundamental elements of a Cloud.

As stated in [6], to deal with a complex system like a Cloud, it is necessary to be able to accurately capture its states.

Beyond the well-known CC characteristics, like on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, etc. [4], [7], it is important to highlight the *stakeholder heterogeneity* characteristic. This characteristic is poorly defined and appears in some works like *stakeholder*, *actors* or *roles*.

In [7] the stakeholders are defined as roles:

- *Cloud Consumer*;
- *Cloud Provider*;
- *Cloud Auditor*;

 ISBN: 978-1-61208-318-6

- *Cloud Broker* ;
- *Cloud Carrier*.

Litoiu at al. [8] presents four type of stakeholders: *infrastructure providers*, *platform providers*, *application providers* and *end users*, although, it does not describe these stakeholders roles. In the same paper [8], there is a change on the terms used to present the stakeholders; the term actors is used instead of stakeholders. It places the actors in function of service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS); introducing a whole different set of actors as *layer owners*, that are: *IaaS owner*, *PaaS owner*, *SaaS owner* and *end users*.

Leimeister et al. [9] defines five actors in the *CC value network*: *Customer*, *Service providers*, *Infrastructure providers*, *Aggregate services providers (aggregators)*, *Platform provider* and *Consulting*.

In [10], it can be observed a different definition of roles on Cloud environments. The work defines these roles as *stakeholders* and present the following concepts: *Consumers*, *Providers*, *Enablers* and *Regulators*.

Letaifa et al. [11] present a definition of actors and roles in cloud computing systems as: *Vendors*, *Developers* and *End users*.

In Tan et al. [12], although the work focus is adoption (or not) of SaaS, it classifies stakeholders in three categories: *SaaS infrastructure provider*, *SaaS provider* and *SaaS consumer*.

There is no concise definition of CC stakeholders and interests. Furthermore, those distinct definitions may indicate that each Cloud implementation requires a specific stakeholder's modeling.

B. Autonomic Computing

The Autonomic Computing (AC) concept was based on the human nervous system, which regulates critical functions such as heart rate and body temperature, in the absence of a conscious brain [13]. AC systems have many common points with Expert Systems (ES) but are less generic, applied to management and control of wide computational systems, while the ES are applied in a more generic way. The AC differs from ES principally when it addresses the "action taking", that was unusual in ESs, as stated in [14].

AC systems are based on MAPE-K control cycle, that consists in *Monitor*, *Analyse*, *Plan*, *Execute* and *Knowledge* elements, Fig. 1 shows the MAPE-K life cycle.

An autonomic system, as shown in [13], to be able to perform *self-management*, must present four main abilities:

- *self-configuration* - the ability of configure itself according to high-level policies;
- *self-optimization* - the capacity of optimize its use of resource;
- *self-protection* - autonomic systems must protect itself from malicious or incorrect user behavior;
- *self-healing* - the ability of detect, diagnoses and fix problems.

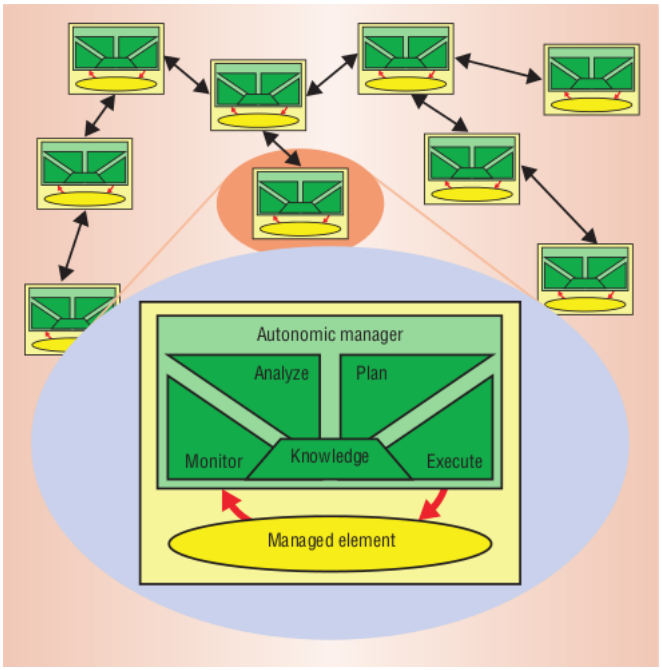


Figure 1. Structure of and autonomic agent.

In [2], the abilities were extended, adding four attributes of autonomic systems:

- *self-awareness* - the system must be aware of its internal state;
- *self-situation* - it should detect its current external operating conditions;
- *self-monitoring* - it has to detect changing circumstances;
- *self-adjustment* - it has to adapt accordingly to external or internal changes.

As stated in [15]:

> The overall goal of Autonomic Computing is the creation of self-managing systems; these are proactive, robust, adaptable and easy to use. Such objectives are achieved though self-protecting, self-configuring, self-healing and self-optimizing activities ... To achieve these objectives a system must be both self-aware and environment-aware, meaning that it must have some concept of the current state of both itself and its operating environment. It must then self-monitor to recognize any change in that state that may require modification (self-adjusting) to meet its overall self-managing goal. In more detail, this means a system having knowledge of its available resources, its components, their desired performance characteristics, their current status, and the status of inter-connections with other systems.

C. Markov Decision Process

Broadly speaking, it can be said that the planning techniques developed in the Artificial Intelligence domain are

 ISBN: 978-1-61208-318-6

concerned to obtain a course of actions which conducts the agent to a goal state or to an improvement in its condition. In deterministic planning approaches, each action leads to a single state. On the other side, the DTP is a non-deterministic way of modeling the decision taken problem where each action (or exogenous event) can lead the system state to more than one possible states with a certain probability.

To deal with probabilistic non-determinism, many mathematical tools must be used. A common framework used as underlying model to DTP is the MDP [16] that exposes the probabilistic relation between the system's states. Another framework is the decision theory [17] which combines the probability theory with utility theory.

In order to model the planning problem for a *stochastic dynamic system*, it is necessary to present a basic problem formulation using a MDP. This paper will model the problem according to [16], that presents the follow key elements:

- a set of decision epochs;
- a set of system state;
- a set of actions;
- a set of transition probabilities (state X action);
- a set of rewards or costs for transitions.

The problem can be mathematically expressed as $\{T, S, A_s, p_t(s'_{t+1}|s,a), r_t(s_t,a)\}$, where S is the set of states that the system can assume; A_s is the set of actions that can be taken over the system at state s; $p_t(s'_{t+1}|s_t,a)$ is the transition function that maps in time t a state s to a state s'_{t+1}, in time $t+1$, give an action a; $r_t(s_t,a)$ is function which gives the reward for the execution of an action a on state s_t.

The Fig. 2 shows a graphical representation of a MDP, where the green circles are the states, the red circles represents the actions, the arrows are the transitions between states, the numbers over the arrow are the probabilities to achieve a state, and the numbers indicated by the yellow arrows are the reward value of the transition.

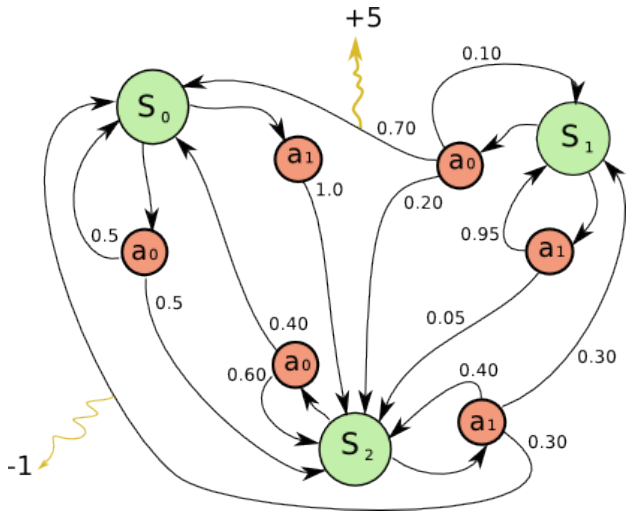


Figure 2. A graphical representation of MDP [18].

III. Related Works

Werner et al. [19] propose an integrated solution for cloud environment based on organization model of autonomous agent. It introduced concepts of rules and beliefs that in this paper can be compared with decision rules and transition functions, respectively. Despite the affinity in terms of goals and some architectural elements, [19] used neither MDP nor decision-theoretic approach and did not deepen the model to a level that would allow some kind of mathematical or logical inference.

The work in [8] presents a model that includes stakeholders, goals, sensors and actuators. It also provides an optimization and control module in the presented architecture which works in a multi-layer approach. However, it did not deepen to a level of inference and did not situate itself on AC context.

In [20], the author describe a MDP focused on the self-configuration phase. It situates the CC management on AC context, applying learning enforcement to deduce the action to be taken. The main differences between [20] and this paper is that the first one maintains the focus just in management of virtualized resource, as well as, not guiding the goals to meet stakeholder's interests.

A multi-level (physical and logical) resource manager is proposed in [1] to self-optimize on AC context. However, the paper uses utility functions regardless the uncertainty inherent to CC. It does not consider the action probabilities in this model.

An introduction of the term Autonomic Cloud Computing was made on [21], however, the work transits just as a survey and architecture proposal, without any inference model.

Sharma et al. [22] consider both utility and probability in their decision model. However, the work just provides a specific model that are based in a static set of actions to be optimized.

IV. Stakeholders, Interests and Cloud Computing

This work introduces the idea of *interests*. Interests have been implicitly referenced in many works that address the CC, like [23]. It is relevant to explain stakeholder's concerns in a way that they lead the decisions on a *cloud autonomic system*.

Sharma et al. [22] presents two approaches on decisions for dynamic provisioning: *cloud provider centric* and *customer-centric*. The paper differentiates them as follows:

> Cloud provider centric approaches attempt to maximize revenue while meeting an applications SLA in the face of fluctuating workloads, while a customer-centric approach attempts to minimize the cost of renting servers while meeting the applications SLA.

This definition states important aspects of decision on CC management:

- different types of stakeholders have different interests over the Cloud;

- any decision method inherently carries the ability to benefit the interests of some stakeholders and harm another;
- the stakeholder's interests are not always reconcilable given certain constraints of resources and service demand;

Therefore, it is possible to postulate that an autonomic system that intends to manage a Cloud must guide its decisions in order to *maximize the total satisfaction of the stakeholders*.

The satisfaction function can have many forms. Here will be considered just a function $\sigma : S \rightarrow \mathbb{R}$, where S is the set of possible *states* of the Cloud, and $\mathbb{R}$ is the set of real numbers.

A. Interests on the Cloud

Any measurable variable may constitute the state of Cloud. The set of variables that define a state $s \in S$ can be something really big. However, looking from the view of the Cloud's stakeholders, the indicators contained in the SLAs, SLOs and in some *operative constraints* can help to reduce the sample space. Therefore, the indicators which will compose a Cloud state must be at most the intersection of variables specified in all SLAs, SLOs and operative constraints. These indicators are the *measurable interests*.

Once the measurable interest that will compose the cloud state set S were selected, it is important to refine the indicators, reducing the amount of information, avoiding redundancy, and adjusting the measures. This way, the states changes would be compatible with the speed of the decision making system.

V. A Decision-Theoretic Model to CC

To introduce a decision-theoretic model in CC, it is necessary to find the elements of MDP on the Cloud.

A. MDP for CC

The first element to be modeled is the decision epochs. The idea of applying MDP to decision making in CC induces to think in discrete time decisions (epochs). For continuous time decisions, control theory have been more adherent.

Decisions on a Cloud environment may be taken in preset time slots or triggered by events coming from the monitoring system. The time slot must be adjusted to be large enough that a selected action can be computed, executed and evaluated before a next decision can be taken. The slot also must be short enough that some important event is not missed. In any case, the decision epochs will be treated as a time instant t.

The second element is the Cloud state, presented in subsection IV-A. It can be defined as a tuple $s = (x_1, x_2, \cdots, x_n)$, where $x_1, x_2, \cdots, x_n$ are values of random variables $X_1, X_2, \cdots, X_n$ that compose the measurable interests set. The set of all possible tuples will be expressed as the set S of the Cloud's states.

The third element is a set of possible actions to be taken in each state $s \in S$, that can be represented as the set $A_s \subseteq A$, where A is the set of all possible actions to be taken over the Cloud in any state.

The fourth item to be modeled is the probability distribution $p_t(s'_{t+1}|s_t, a)$, that express the probability of the system to assume the state $s'_{t+1} \in S$ at time $t+1$ giving that was executed the action $a \in A_s$ at state s_t in time t.

At last, the fifth element is a real-valued reward function $r_t : S \times A \rightarrow \mathbb{R}$ (in another form: $r_t(s_t, a)$) that gives the reward received by the decision maker at time t for the execution of the action a in state s_t.

B. Histories, Decision Rules and Policies

Considering that at time 1 an action a_1 executed over a state s_1 will generate a state s_2 at time 2, it is plausible to say that after a time t there will exist a history $h_t = (s_1, a_1, s_2, a_2, \cdots, s_t)$. Being the H_t the set of all possible histories, that will be characterized by the Cartesian product $\{S \times A \times S \times A \times \cdots \times A \times S\} = \{S \times A\}^{t-1} \times S$, it is possible to say that the history h_t constitute an *observation* of system states and actions.

A decision rule for CC can not be memory less. Likewise, must be history dependent. Also, in a real Cloud management, the rule should be *non stationary*, in other words, itself will change over time. Therefore, the decision function to the epoch t ($d_t : H_t \rightarrow A_{s_t}$) when receiving a decision history h_t returns an action a. In another form: $d_t(h_t) \in A_{s_t}$.

The specific decision rules implemented will be described in the Section VI.

As depicted in Section II, a policy provides to decision maker, a prescription for an action selection under any possible future system state. Thus, on a Cloud where the decision rules are time and history dependent, there will a policy which will look like: $\pi = (d_1, d_2, d_3, \cdots, d_t)$.

Clearly, when the decision horizon is infinite (as in CC), it is impractical to compute and evaluate all decision possibilities for all future states that the Cloud may assume in time. Although it has a finite horizon for planning, the size of S and A may result in a really hard work, since the number of policies is $\{S \times A\}^{N-1} \times S$, where N is the time horizon to compute.

VI. Mathematical Modeling

A. Study Case: the Cloud

1) Cloud environment: The Cloud used to ground the experiments was a CloudStack [3], using Xen Cloud Platform [24] and Xen hypervisor [25], [26] at the bare layer.

The Cloud Lab can be summarized as follows:

- Top level, cloud management – applying Cloudstack over the infrastructure to manage not just the hardware and software pieced together, but also to provide an easy and user friendly way to create, destroy and update resources on the fly;
- Underlying structure, hypervisor level – the core structure that is responsible for running the VMs. It is based on Ubuntu server 12.10 64bits, on which was installed and configured the Xen hypervisor 64bits hypervisor version 4.1 and Xen Cloud Plataform (XCP) version

 ISBN: 978-1-61208-318-6

TABLE I. CLOUD SERVERS CONFIGURATIONS

Name	CPU	Memory	Role	Cluster
Server 1	Pentium D @ 3.4Ghz	2 GB	Storage server	-
Server 2	Core 2 E6600 @ 2.4Ghz	7GB	Processing server	PV
Server 3	Core 2 quad, Q8200 @2.3Ghz	4GB	Processing server	PV
Server 4	Xeon E5405 @ 2Ghz	8GB	Processing server	HVM
Server 5	Phenom 9650 @ 2.3Ghz	4GB	Processing server	HVM
Server 2	Phenom II 965 @ 3.4Ghz	4GB	Processing server	HVM

1.6. Since the platform has a heterogeneous server environment, this level was then subdivided into two clusters, a cluster that contains the physical hosts with hardware virtual machine capabilities (HVM) and a pure virtual cluster (PV) that contains all the servers that does not have support for HVM machines;

- Bottom level structure, storage level – at the lower level of the structure there is the storage, on which was built upon a Raid controller. The Raid controller exports 3 volumes to the storage server, a total of 3.18 Terabits using either RAID1 or RAID5, granting at least reliability against hard drive failures. The volumes exported by the RAID controller are mounted on a passive storage server as XFS file system and then exported the XFS partitions as network file system (NFS).

Table I shows the hardware configurations of each server in the Cloud, and the role that it plays.

2) Cloud Stakeholders: There are three types of stakeholders interested in the Cloud environment:

- the *cloud decision committee* – that is interested in the maximization of resource usage and the equipment aggregation of research departments by accession to Cloud, in the scope of the Federal University of Santa Catarina, Brazil;
- the *cloud managers* – which have interest in satisfying the concerns of cloud decision committee with availability assurance and energy economy;
- the *researchers* – having interests that their VMs stay available and maximize their capacity of memory, CPU and storage.

3) Cloud problems: From the stakeholders interests it is possible derive some decision problems:

1) Structure Aggregation: It is necessary to decide if a PM can integrate the cloud or not;
2) Maximize the resource usage: It is necessary to add and remove resource (CPU, memory) to idle and overloaded VMs;
3) Ensure availability: It is necessary not to lose user requests, that can be characterized as network packages.
4) Energy Economy: It is necessary to minimize the energy consumption;
5) Obtain availability and maximize the capacities of VMs: the decision needs are covered on items 3 and 2.

These decision problems have consequences when placed side-by-side. Although it is possible to provide more resource to VMs via hypervisor, the operation system will not recognize the additional resource, to add or remove them when it is necessary stop and restart the VMs. This way, to provide availability, it is critical to execute these operations in a moment with low probability of request loss.

To achieve energy economy, it is important to shutdown PMs, but, it is necessary to execute VM migrations at a moment when the request loss is not likely to happen. The aggregation of new equipments can be made at any time if there exists a good link quality between the new host and the core of the cluster where it will be added.

B. The model

To get a mathematical modeling that covers the study case, it is necessary to model four basic structures, which will be presented below.

1) Cloud State: In order to cover stakeholders interests, there was selected a set of basic metrics to compose the Cloud states as follows:

- the PM state – on or off – $S^{PM} = \{on, \overline{on}\}$;
- the PM energy consumption – level of consumption – $S^{E} = \{low, average, high\}$;
- VM state – up or down – $S^{VM} = \{up, \overline{up}\}$;
- CPU – level of CPU utilization – $S^{cpu} = \{idle, underused, welldimensioned, overloaded\}$;
- memory – level of memory utilization – $S^{mem} = \{idle, underused, welldimensioned, overloaded\}$;
- network link utilization– level of link utilization – $S^{lu} = \{idle, underused, wellused, overloaded\}$;
- network link quality – level of link quality – $S^{lq} = \{poor, regular, good\}$;

Given the sets PM of all PMs in the Cloud, VM of all created VMs, it is possible to define the following sets to compose the Cloud state:

- $ST^{PM} = PM \times S^{PM}$, of each PM state;
- $SE^{PM} = PM \times S^{E}$, of each PM energy consumption;
- $SC^{PM} = PM \times S^{cpu}$, of PM CPU utilization;
- $SM^{PM} = PM \times S^{mem}$, of PM memory utilization;
- $LU^{PM} = PM \times S^{lu}$, of PM network connection utilization;
- $LQ^{PM} = PM \times S^{lq}$, of PM network link quality;
- $ST^{VM} = VM \times S^{PM}$, of each VM state;
- $SC^{VM} = VM \times S^{cpu}$, of VM CPU usage;
- $SM^{VM} = VM \times S^{mem}$, of VM memory utilization;
- $LU^{VM} = VM \times S^{lu}$, of VM network connection utilization;
- N^{PM}, the number of PMs in the Cloud.

 ISBN: 978-1-61208-318-6

This way, there exists a set that can cover all possible Cloud state is the set $S = ST^{PM} \times ST^{E} \times SC^{PM} \times SM^{PM} \times LU^{PM} \times LQ^{PM} \times ST^{VM} \times SC^{VM} \times SM^{VM} \times LU^{VM} \times N^{PM}$, where a state can be represented at time t as $s_t \in S$. The set of states in this example cannot be stationary, once VMs may be created or destroyed and PMs can be added or removed.

2) Actions: It is the set of base actions that can be executed on the environment that is being used to run the experiments:

- a_1 – turn on PM;
- a_2 – turn off PM;
- a_3 – turn on VM;
- a_4 – turn off VM;
- a_5 – migrate VM;
- a_6 – scale up VM;
- a_7 – scale down VM;
- a_8 – admit PM;
- a_9 – refuse PM;
- a_0 – no action;

It will be considered as different action, the actions executed on distinct resource (e.g., $a_1(pm_1)$). This way, the set of action will be larger than the nine basic actions listed above. However, the actions cannot be executed in any state of a resource. For instance, a PM in $\overline{up}$ state cannot be turned off. To get the set of all possible actions which can be performed in state s_t, there will be defined a function $\alpha_t : S \rightarrow 2^A$ that receives the state of the Cloud and returns a set of actions, where 2^A is the power set of A set. In another form $\alpha_t(s_t) = A_{s_t}$. As the Cloud state, α is not a stationary functions and can change in time.

3) Transition Function: Considering a state $s_t \supset \{ST^{PM_1} = on\}$ over which the action $a_2^{PM_1}$ is executed, it will be induced to a state $s'_{t+1} \supset \{ST^{PM_1} = off \cup LQ^{PM_1} = bad\}$ with a probability x, if there is demand to PM_1, and a state $s''_{t+1} \supset \{ST^{PM_1} = off \cup LQ^{PM_1} = \{regular, good\}\}$ with probability $1 - x$ otherwise, depending on whether the machine is on or off, and how much requests exits to its VMs.

The state and action relations must be captured by the transition function. It must provide the probability that an action a, executed on state s_t, results in a state s'_{t+1}, for each time t. In other words, $p_t(s'_{t+1}|a, s_t)$.

On a real-world CC management application, the transition functions will be the product of a series of machine learning and forecasting methods. In this paper, we will only assume that there is a non-stationary probability function.

4) Reward Function: Giving the satisfaction evaluation function described in Section IV, here presented as $v = \sigma(s_t)$, it is possible to establish a reward function that evaluates the decision maker reward to lead the system from a state s_t to a state s_{t+1}. It can be expressed as: $r_t(s_t, a, s_{t+1}) = \sigma(s_{t+1}) - \sigma(s_t)$.

When a decision maker recognizes the Cloud in state s_t and evaluates the impact of action a over that state, it does not have the resultant state s_{t+1}. This way, it is necessary to introduce the uncertainty, as seen on (1).

$$r_t(s_t, a) = \sum_{s_{t+1} \in S} r(s_t, a, s_{t+1}).p_t(s_{t+1}|s, a) \quad (1)$$

It is assumed that each stakeholder $k \in K$ has a set of interests $I_k \subseteq I$, each $i_k \in I_k$ having a weight w_{i_k}. Therefore, if there exists a function $\sigma^{I_k} : S \rightarrow \mathbb{R}$ that evaluates the Cloud state according to the interest i of stakeholder k, it is possible to propose a weighted interest function σ^k as presented in (2).

$$\sigma^k = \sum_{i_k \in I^k} w_{i_k} \cdot \sigma^{I_k}(s_t) \quad (2)$$

The σ function presented above can be obtained by the sum of all weighted interests, as presented by (3).

$$\sigma = \sum_{k \in k} \sigma^k(s_t) \quad (3)$$

The expected reward is computed as presented in (1).

C. Decision rules

It is simple to elaborate a decision rule for one MDP epoch. All is needed is the selection of the action that has the best expected reward (like in (1)). This way, the decision rule to $t = 1$ must return the set of the best actions as shown in (4).

$$A_t^* = \arg\max_{a \in A} \{ \sum_{s_{t+1} \in S} r(s_t, a, s_{t+1}).p_t(s_{t+1}|s_t, a) \} \quad (4)$$

Nevertheless, to compute the expected reward for a policy π_t^N, where t is the actual time and N is the horizon of the policy, it is necessary to sum the product of all rewards that achieve a final state and the probability to achieve each state, like in (5).

$$r(\pi_t^N) = \sum_{h_t^N \in H_t^N} (\sigma(s_N) - \sigma(s_t)) \cdot p(s_N) \quad (5)$$

The $p(s_N)$ can be computed from the sum of all probabilities of histories $p(h_t^N)$ that start on s_t and finishes on s_N. Considering $X_N(h_t^N) = s_N$ as a random variable that returns the state N of a history, it is possible observe in (6) how to compute a state probability.

$$p(s_x) = \sum_{h_t^N, X_N = s_x} p(h_t^N) \quad (6)$$

The probability of a history can be calculated by (7).

$$p(h_t^N) = p(s_{t+1}|s_t, a_t) \times \prod_{i=t+1}^{N-1} p(s_{i+1}|s_i, a_i) \quad (7)$$

 ISBN: 978-1-61208-318-6

VII. Conclusion and Future Works

This article has presented a decision-theoretic modeling to decision making for CC management in an AC context, using the MDP as a mathematical framework.

This paper contributed to the state-of-the-art in CC research in the sense that it tackles the phase *planning* of an autonomic cycle with a mathematical model which takes into consideration the uncertainty of action resulting in complex systems such as a CC management systems.

For future work, the following steps will be considered:

- A big data model to feed the transition function created with the monitoring data bases;
- Extend the CloudStack to implement the model on its resource manager and perform experiments to observe the performance of the model in taking decisions;
- Analyze a meta-management model to optimize the autonomic planner;
- Research methods of action discovery and learning.

References

[1] W. Walsh, and G. Tesauro, and J. Kephart, and R. Das, "Utility functions in autonomic systems", in: Autonomic Computing, 2004. Proceedings. International Conference on, May 2004, pp. 70–77.

[2] S. Dobson, and R. Sterritt, and P. Nixon, and M. Hinchey, "Fulfilling the vision of autonomic computing", Computer, vol. 43, no. 1, Jan. 2010, pp. 35–41.

[3] Apache Foundation. Apache CloudStack, 2013 retrieved in September 2013 from http://cloudstack.apache.org/

[4] P. Mell and T. Grance, The NIST Definition of Cloud Computing, Tech. rep., National Institute of Standards and Technology, Information Technology Laboratory, Jul. 2009.

[5] G. Aceto, and A. Botta, and W. de Donato, and A. Pescapè, "Cloud monitoring: A survey", Computer Networks, vol. 57, issue 9, Jun. 2013, pp. 2093–2115.

[6] A. Viratanapanu, and A. Hamid, and Y. Kawahara, and T. Asami, "On demand fine grain resource monitoring system for server consolidation", Kaleidoscope: Beyond the Internet? - Innovations for Future Networks and Services, 2010 ITU-T, Dec. 2010, pp. 1–8.

[7] R. B. Bohn, and J. Messina, and F. Liu, and J. Tong, and J. Mao, "Nist cloud computing reference architecture", in: Services (SERVICES), 2011 IEEE World Congress on, July 2011, pp. 594–596.

[8] M. Litoiu, and M. Woodside, and J. Wong, and J. Ng, and G. Iszlai, "A business driven cloud optimization architecture", in: In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10), 2010, pp. 380–385.

[9] S. Leimeister, and M. Bhm, and C. Riedl, and H. Krcmar, "The business perspective of cloud computing: Actors, roles and value networks", in: In Proceedings of the 7th international conference on Economics of grids, clouds, systems, and services (GECON'10), 2010, pp. 129–140.

[10] S. Marston, and Z. Li, and S. Bandyopadhyay, and J. Zhang, and A. Ghalsasi, "Cloud computing - The business perspective". Decis. Support Syst. vol. 51, no. 1, Apr. 2011, pp. 176–189.

[11] A. B. Letaifa, and A. Haji, and M. Jebalia, and S. Tabbane, "State of the Art and Research Challenges of new services architecture technologies: Virtualization, SOA and Cloud Computing", International Journal of Grid & Distributed Computing, vol. 3, issue 4, Dec. 2010, pp. 69–88.

[12] C. Tan, and K. Liu, and L. Sun, "A design of evaluation method for saas in cloud computing", Journal of Industrial Engineering and Management, vol. 6, no. 1, Feb. 2013, pp. 50–72.

[13] J. Kephart and D. Chess, "The vision of autonomic computing", Computer, vol. 36, no. 1, Jan. 2003, pp. 41–50.

[14] S. Gutirrez and J. Branch, "A comparison between expert systems and autonomic computing plus mobile agent approaches for fault management", DYNA, vol. 78, no. 168, Aug. 2011, pp 173–180.

[15] R. Sterritt and D. Bustard, "Autonomic computing - a means of achieving dependability?", in: Engineering of Computer-Based Systems, 2003. Proceedings. 10th IEEE International Conference and Workshop on the, Apr. 2003, pp. 247–251.

[16] M. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley Series in Probability and Statistics, Wiley, 2009.

[17] B. Lindgren, Elements of decision theory, Macmillan, 1971.

[18] Wikipedia, Markov Decision Process, 2013 retrieved in September 2013 from http://en.wikipedia.org/wiki/Markov_decision_process

[19] J. Werner, and G. Geronimo, and C. B. Westphall, and F. L. Koch, and R. Freitas, C. M. Westphall, "Environment, services and network management for green clouds", CLEI Electron. J. vol. 15, no. 2, Aug. 2012, pp 2–2.

[20] J. Rao, Autonomic Management of Virtualized Resources in Cloud Computing, Ph.D. thesis, Wayne State University, Jan. 2011.

[21] R. Buyya, and R. Calheiros, and X. Li, "Autonomic cloud computing: Open challenges and architectural elements", in: Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on, Sep. 2012, pp. 3–10.

[22] U. Sharma, Elastic resource management in cloud computing platforms, Ph.D. thesis, University of Massachusetts, May 2013.

[23] D. Durkee, "Why Cloud Computing Will Never Be Free", Queue Journal vol. 8 no. 4, Apr. 2010, pp. 20-29.

[24] The Linux Foundation, Xen Cloud Platform, June 2013 retrieved in September 2013 from http://www.xenproject.org/downloads/xen-cloud-platform-archives.html

[25] The Linux Foundation, Xen Hypervisor, 2013 retrieved in September 2013 from http://www.xenproject.org/users/virtualization.html

[26] P. Barham et al., "Xen and the Art of Virtualization", in: Proceedings of the nineteenth ACM symposium on Operating systems principles, Oct. 2003, pp. 164–177.

 ISBN: 978-1-61208-318-6

Prioritized Adaptive Max-Min Fair Residual Bandwidth Allocation for Software-Defined Data Center Networks

Andrew Lester
School of Information Technology
Illinois State University
aeleste@ilstu.edu

Yongning Tang
School of Information Technology
Illinois State University
ytang@ilstu.edu

Tibor Gyires
School of Information Technology
Illinois State University
tbgyires@ilstu.edu

***Abstract*—Modern data center networks commonly adopt multi-rooted tree topologies. Equal-Cost Multi-Path (ECMP) forwarding is often used to achieve high link utilization and improve network throughput. Meanwhile, max-min fairness is widely used to allocate network bandwidth fairly among multiple applications. However, today's data centers usually host diverse applications, which have various priorities (e.g., mission critical applications) and service level agreements (e.g., high throughput). It is unclear how to adopt ECMP forwarding and max-min fairness in the presence of such requirements. We propose Prioritized Max-Min Fair Multiple Path forwarding (PMP) to tackle this challenge. PMP can optimally allocate current available bandwidth to maximally satisfy user demands. When predefined application requirements are available, PMP can prioritize current demands and allocate available bandwidth accordingly. Our performance evaluation results show that PMP can improve application throughput 10-12% on average and increase overall link utilization especially when the total demanded bandwidth is close or even exceeds the bisectional bandwidth of a data center network.**

Keywords- SDN, max-min fair, scheduling.

I. INTRODUCTION

In recent years, data centers have become critical components for many large organizations [1] [2] [19]. A data center (DC) refers to any large, dedicated cluster of computers that is owned and operated by a single authority, built and employed for a diverse set of purposes. Large universities and private enterprises are increasingly consolidating their IT services within on-site data centers containing a few hundred to a few thousand servers. On the other hand, large online service providers, such as Google, Microsoft, and Amazon, are rapidly building geographically diverse cloud data centers, often containing more than 10,000 servers, to offer a variety of cloud-based services such as web servers, storage, search, on-line gaming. These service providers also employ some of their data centers to run large-scale data-intensive tasks, such as indexing Web pages or analyzing large data-sets, often using variations of the MapReduce paradigm.

Many data center applications (e.g., scientific computing, web search, MapReduce) require substantial bandwidth. With the growth of bandwidth demands for running various user applications, data centers also continuously scale the capacity of the network fabric for new all-to-all communication patterns, which presents a particular challenge for traditional data forwarding (switching and routing) mechanisms. For example, MapReduce based applications, as a currently adopted default computing paradigm for big data, need to perform significant data shuffling to transport the output of its map phase before proceeding with its reduce phase. Recent study shows the principle bottleneck in large-scale clusters is often inter-node communication bandwidth. Traffic pattern study [17] showed that only a subset (25% or less) of the core links often experience high utilization.

Modern data center networks commonly adopt multi-rooted tree topologies [1] [2] [19]. ECMP is often used to achieve high link utilization and improve network throughput. Meanwhile, max-min fairness is widely used to allocate network bandwidth fairly among multiple applications. Many current data center schedulers, including Hadoops Fair Scheduler [26] and Capacity Scheduler [25], Seawall [24], and DRF [27], provide max-min fairness. The attractiveness of max-min fairness stems from its generality. However, today's data centers usually host diverse applications, which have various priorities (e.g., mission critical applications) and service level agreements (e.g., high throughput). It is unclear how to adopt ECMP forwarding and max-min fairness in the presence of such requirements. We propose Prioritized Max-Min Fair Multiple Path forwarding (PMP) to tackle this challenge. PMP can optimally allocate current available bandwidth to maximally satisfy user demands. When predefined application requirements are available, PMP can prioritize current demands and allocate available bandwidth accordingly.

The disruptive Software-Defined Networking (SDN) technology shifts today's networks that controlled by a set of vendor specific network primitives to a new network paradigm empowered by new programmatic abstraction. OpenFlow provides a protocol such that the logical centralized controller can exploit forwarding tables on SDN switches for programmatic multi-layer forwarding flexibility. One of the fundamental transformations that flow based forwarding presents is the inclusion of multi-layer header information to make forwarding match and action logic pro-

 ISBN: 978-1-61208-318-6

grammatically. Programmatic policy is vital to manage the enormous combinations of user requirements. For example, an SDN controller can flexibly define a network flow using a tuple as (incoming port, MAC Src, MAC Dst, Eth Type, VLAN ID, IP Src, IP Dst, Port Src, Port Dst, Action).

The rest of the paper is organized as the following. Section II discusses the related research work. Section III formalizes the problem and describes our approach. Section IV presents our simulation design and results, respectively. Finally, Section V concludes the paper with future directions.

II. Related Work

Current large data center networks connect multiple Ethernet LANs using IP routers and run scalable routing algorithms over a number of IP routers. These layer 3 routing algorithms allow for shortest path and ECMP routing, which provide much more usable bandwidth than Ethernets spanning tree. However, the mixed layer 2 and layer 3 solutions require significant manual configuration.

The trend in recent works to address these problems is to introduce special hardware and topologies. For example, PortLand [2] is implementable on Fat Tree topologies and requires ECMP hardware that is not available on every Ethernet switch. TRILL [3] introduces a new packet header format and thus requires new hardware and/or firmware features.

There have been many recent proposals for scale-out multi-path data center topologies, such as Clos networks [4], [5], direct networks like HyperX [6], Flattened Butterfly [8], DragonFly [8]. etc.,, and even randomly connected topologies have been proposed in Jellyfish [9].

Many current proposals use ECMP-based techniques, which are inadequate to utilize all paths, or to dynamically load balance traffic. Routing proposals for these networks are limited to shortest path routing (or K-shortest path routing with Jellyfish) and end up underutilizing the network, more so in the presence of failures. While DAL routing [6] allows deroutes, it is limited to HyperX topologies. In contrast, Dahu [19] proposes a topology-independent, deployable solution for non-minimal routing that eliminates routing loops, routes around failures, and achieves high network utilization.

Hedera [10] and MicroTE [13] propose a centralized controller to schedule long lived flows on globally optimal paths. However they operate on longer time scales and scaling them to large networks with many flows is challenging. Techniques like Hedera, which select a path for a flow based on current network conditions, suffer from a common problem: when network conditions change over time the selected path may no longer be the optimal one. While DevoFlow [14] improves the scalability through switch hardware changes, it does not support non-minimal routing or dynamic hashing. Dahu can co-exist with such techniques to better handle congestion at finer time scales.

MPTCP [11] proposes a host based approach for multi-path load balancing by splitting a flow into multiple sub flows and modulating how much data is sent over different subflows based on congestion. However, as a transport protocol, it does not have control over the network paths taken by subflows. Dahu [19] exposes the path diversity to MPTCP and enables MPTCP to efficiently utilize the non-shortest paths in a direct connect network. There have also been proposals that employ variants of switch-local per-packet traffic splitting [20].

Traffic engineering has been well studied in the context of wide area networks. TeXCP [21], and REPLEX [22] split flows on different paths based on load. However, their long control loops make them inapplicable in the data center context that requires faster response times to deal with short flows and dynamic traffic changes. FLARE [7] exploits the inherent burstiness in TCP flows to schedule "flowlets" (bursts of packets) on different paths to reduce extensive packet reordering.

III. Prioritized Adaptive Max-Min Fair Bandwidth Allocation

While ECMP is often used to achieve high link utilization, max-min fairness is widely used to allocate network bandwidth fairly among multiple applications. However, today's data centers usually host diverse applications, which have various priorities (e.g., mission critical applications) and service level agreements (e.g., high throughput). It is unclear how to adopt ECMP forwarding and max-min fairness in the presence of such requirements. We propose Prioritized Max-Min Fair Multiple Path forwarding (PMP) to tackle this challenge. In the following, we first formalize the problem, and then present how PMP works.

A. Problem Formalization

Consider a data center network with K-ary fat-tree topology as shown in Fig.1, composed of a set of core switches S_c, a set of aggregation switches S_a, a set of edge switches S_e, and a set of hosts H. Each switch has k-port. There are k pods. Each pod contains $k/2$ aggregation switches and $k/2$ edge switches. In each pod, each k-port edge switch is directly connected to $k/2$ hosts and $k/2$ aggregation switches. The i^{th} port of each core switch $s_i \in S_c (i \in [1, (k/2)^2])$ is connected to pod i [2]. We assume all links (e.g., L_1 in Fig.1) have the same bandwidth for both uplink (e.g., L_1^u) and downlink (e.g., L_1^d) connections.

Recent study [17] showed that less than 25% of the core links have been highly utilized while packet losses and congestions may still often occur. In this paper, we only focus on inter-pod network traffic that requires bandwidth from core links. We denote all links between aggregation and core layers as a set L_{ac}, all links between edge and aggregation layers as a set L_{ea}, and all links between application server and edge layers as a set L_{se}. Generally,

 ISBN: 978-1-61208-318-6

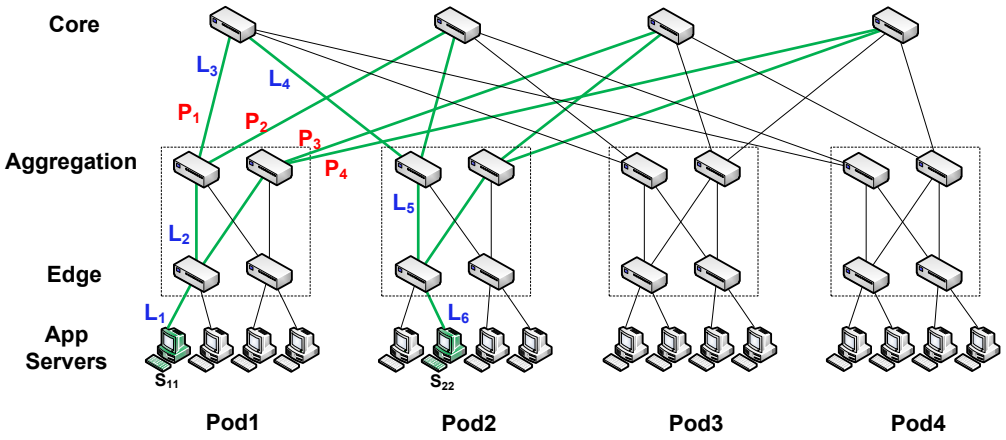


Figure 1: Fat tree topology.

in a network with K-ary fat-tree topology , there are k paths between any two hosts from different pods.

A network task T_i is specified by a source and destination hosts (e.g., S_{11} and S_{22}) and the expected traffic volume. We also consider each task with different priority level w_i. Here, $w_i \in [1, m]$ with the lowest and highest priority levels as 1 and m, respectively. A network scheduler modular (also simply referred to as scheduler) on an SDN controller needs to decide how to allocate available bandwidth to maximally satisfy the application requirements. We define a valid Network Path Assignment PA_i for a given task T_i is a set of paths and their corresponding allocated bandwidths connecting the source to the destination (e.g., a subset of $\{P_1, P_2, P_3, P_4\}$), in which each path consists of a list of directional links (e.g., $P_1 = \{L_1^u, L_2^u, L_3^u, L_4^d, L_5^d, L_6^d\}$) connecting the source to the destination hosts. Here, $L_1^u, L_6^d \in L_{se}$; $L_2^u, L_5^d \in L_{ea}$; $L_3^u, L_4^d \in L_{ac}$.

There is a variety of applications on a data center network, which have different service requirements regarding throughput, packet loss, and delay. For our analysis, we characterize the applications' requirements through their priority levels, which can be the output of some utility function. Priorities can offer a basis for providing application and business oriented service to users with diverse requirements. We consider a model where the weight associated with the different priority classes is user-definable and static. Users can freely define the priority of their traffic, but are charged accordingly by the network. We aim to study the bandwidth-sharing properties of this priority scheme. Given a set of network tasks $T = \{T_i\}$ $(i \geq 1)$ and their corresponding priority levels $K = \{K_i\}$, we consider a Network Path Assignment problem is to find a set of path assignment $PA = \{PA_i\}$ to satisfy the condition of Prioritized Max-Min Fairness.

Definition 1. Prioritized Max-Min Fairness A feasible path assignment PA^x is "prioritized max-min fair" if and only if an increase of any path bandwidth within the domain of feasible bandwidth allocations must be at the cost of a decrease of some already less allocated bandwidth from the tasks with the same or higher priority level. Formally, for any other feasible bandwidth allocation scheme PA^y, if $BW(PA^y_{T_i}) > BW(PA^x_{T_i})$, then it decreases the allocated bandwidth of some other path with the same or higher priority level. Here, $BW(PA^y_{T_i})$ is the total allocated bandwidth for the task T_i in the bandwidth allocation scheme PA^y.

Definition 2. Saturated Path A path P_i is saturated if at least one bottleneck link L_j exists on the path P_i. A link is bottlenecked if the total assigned bandwidth on this link from the given tasks is more than or equal to the maximum bandwidth of the link. Formally, a bottleneck link is the one that $\sum_i BW_{T_i}(L_j) \geq BW_{max}(L_j)$.

B. The Algorithm of Multi-Level Progressive Filling

The network tasks can be dynamically and continuously generated, and submitted to the scheduler. In PMP, the scheduler can periodically query all network switches to collect current link utilizations. Once a new task list received, the scheduler will use a practical approach called "progressive filling" [23] provisioning available bandwidth that results in a prioritized max-min fair allocation following the priority order from the highest to the lowest priority level. The idea is shown in Fig. 2: The scheduler starts with all provisioned bandwidth equal to 0 and increases all bandwidths together at the same pace for the tasks with the same priority level until one or several saturated paths are found. The bandwidth for the corresponding tasks that use these paths are not increased any more and the scheduler continue increasing the bandwidth for other tasks on the same priority level. All the tasks that are stopped have a saturated path. The algorithm continues until it is not possible to increase the bandwidth for the tasks at certain priority level. Then, the algorithm moves to the next

 ISBN: 978-1-61208-318-6

Input: A list of tasks $\{T_i\}$; current link utilization $U(L_j)$
Output: Path assignment PA with PA_i for each task T_i

1: Sort $\{T_i\}$ based on their priority levels K_i
2: Start from the highest priority $W = m$ /*m is the highest priority level*/
3: **for all** $T_i! = \emptyset$ $(PL(T_i) = W)$ **do**
4: /*The function $PL()$ returns the priority level of a given task*/
5: Find all paths for each task T_i
6: Assign a unit bandwidth (UB) to the least utilized path for each task /*we choose UB = 100Kbps*/
7: $PA_i \leftarrow \{T_i, \{P_i\}\}$
8: $PA \leftarrow PA \cup \{PA_i\}$
9: **if** A path P is saturated and $P \in APL(T_i)$ **then**
10: $APL(T_i) \leftarrow APL(T_i) - P$
11: **end if**
12: **if** $APL(T_i) == \emptyset$ **then**
13: Remove T_i
14: **end if**
15: **if** $(\{T_i\} == \emptyset)$ and $(W > 1)$ **then**
16: $W = m - 1$
17: **end if**
18: **end for**
19: return PA

Figure 2: Multi-Level Progressive Filling Algorithm

priority level and repeats the same bandwidth provisioning operations until all tasks are assigned to some paths. The algorithm terminates because the total paths and tasks are finite. When the algorithm terminates all tasks have been served at some time and thus have a saturated path. By Definition 1 the allocation is max-min fair for the tasks at the same priority level.

IV. Evaluation

A. Data Center Network Traffic Pattern

Several recent studies [16]–[18] have been conducted in various data center networks to understand network traffic patterns. The studied data center networks include university campus, private enterprise data centers, and cloud data centers running Web services, customer-facing applications, and intensive Map-Reduce jobs. The studies have shown some interesting facts: (1) The majority of the traffic in data center networks is TCP flows. (2) Most of the server generated traffic in the cloud data centers stays within a rack, while the opposite is true for campus data centers. (3) At the edge and aggregation layers, link utilizations are fairly low and show little variation. In contrast, link utilizations at the core network are high with significant variations over the course of a day. (4) In some data centers, a small but significant fraction of core links appear to be persistently congested, but there is enough spare capacity in the core to alleviate congestion. (5) Losses on the links that are lightly utilized on the average can be attributed to the bursty nature of the underlying applications run within the data centers.

B. Methodology and Metrics

In our experiments, we simulate a data center with a fat-tree topology. We implemented PMP based on RipL [28], a Python library that simplifies the creation of data center code, such as OpenFlow network controllers, simulations, or Mininet topologies. We compared PMP scheduler with a commonly used randomization based scheduling method.

In our evaluation, we use three different priority policies for a mixture of traffic patterns: (1) high priority for long TCP flows with the total data size between $1MB$ and $100MB$; (2) high priority for short TCP flows with the total data size between $10KB$ and $1MB$; (3) high priority for random selected flows including both short and long ones referred to as mixed TCP flows.

We focus on two performance metrics: (1) Link Utilization that demonstrates how effectively the scheduler utilizes the network bandwidth. Intuitively, when there are high bandwidth demands from user applications, the overall link and path utilizations should be kept in high. (2) Network throughput that shows how efficiently the network serves different applications.

C. Link Utilization

We created 16 test scenarios to evaluate PMP with different inter-pod traffic patterns. We ran 5 tests for each scenarios. In all test scenarios, the test traffic traversed all edge, aggregation, and core links. The results of multiple test runs from the same test scenario present similar results. In the following, we only report the result of one test run for each test scenario that created traffic between two pods in both directions. Under the same three different priority policies, Fig.3(a)$\sim$(c) shows the overall path utilization; Fig.4(a)$\sim$(c) shows the aggregation link utilizations; and Fig.5(a)$\sim$(c) shows the core link utilizations. Comparing to the randomization based scheduler, our algorithm 2 achieves high utilization on path level, aggregation and core link levels by: (1) dynamically observing all link utilization status, and (2) progressively filling the jobs of the same priority with the available bandwidth with the max-min fairness. The average gain on utilization is approximately improved from 59% to 66%. Note that with the increase of link utilization, idle bandwidth can be effectively utilized by demanding network applications, which can correspondingly improve their performance by reducing their network latencies.

D. Network Throughput

Once the overall utilization can be increased, we expect that the overall application throughput should also be improved. The experiment results presented some interesting

 ISBN: 978-1-61208-318-6

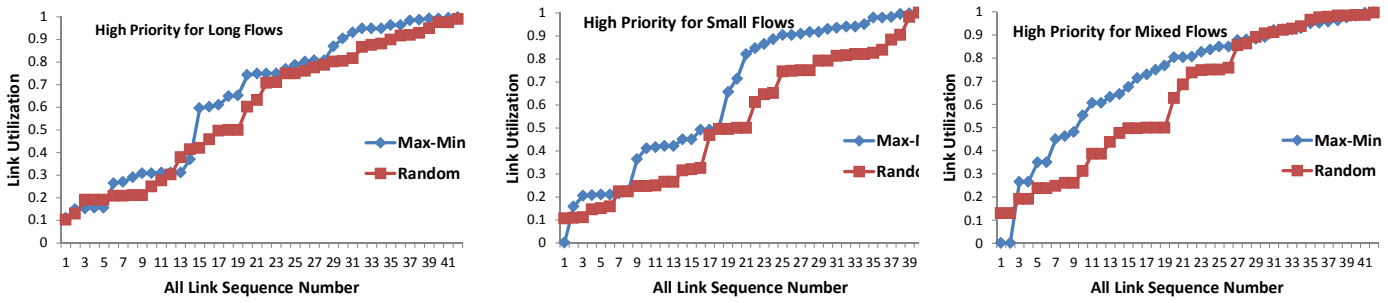


Figure 3: Path utilization with high priority for (a) long flows (b)short flows (c)mixed flows

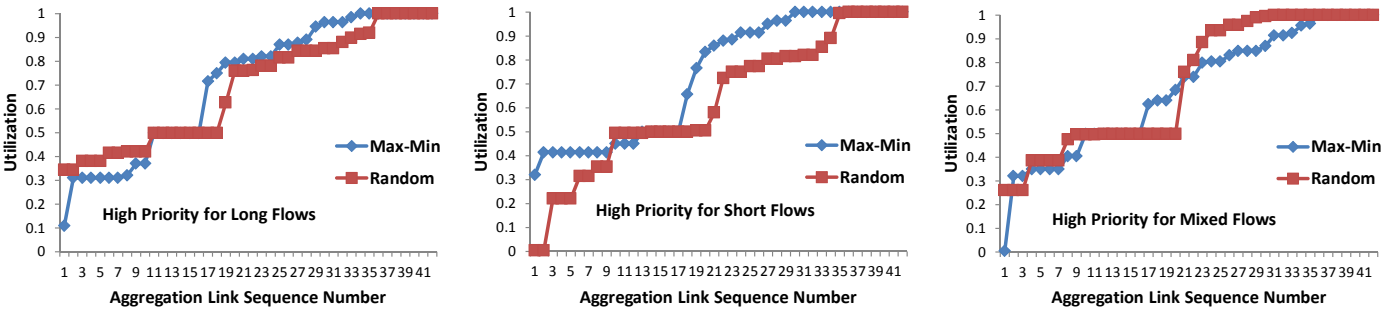


Figure 4: Aggregation link utilization with high priority for (a) long flows (b)short flows (c)mixed flows

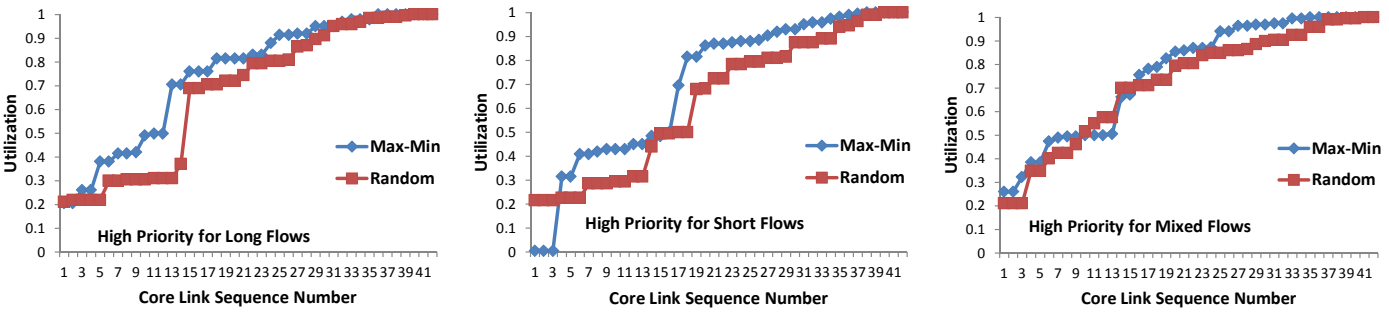


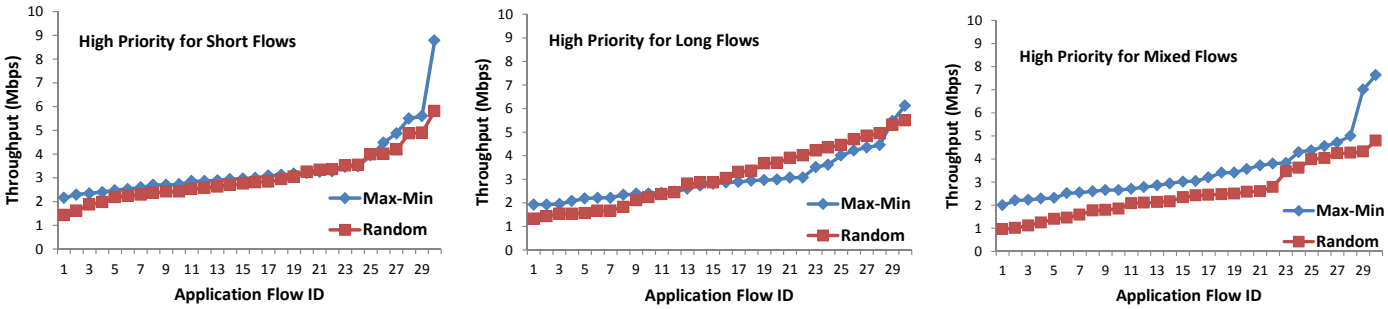
Figure 5: Core link utilization with high priority for (a) long flows (b)short flows (c)mixed flows

results as shown in Fig.6(a)∼(c). When we emulate more realistic application scenarios, where short and long TCP flows are randomly mixed together, our PMP scheduler obviously outperforms the performance of the random scheduler with about 10-12% improvement. In the scenario of the different policies favoring either short or long flows, our scheduler adopts max-min fairness, and thus, the average throughput has been improved from $2.52Mbps$ in the random scheduler and to $3.46Mbps$ in the max-min scheduler.

V. Conclusion

The role of the data center network is becoming ever more crucial today, which is evolving into the integrated platform for next-generation data centers. Because it is pervasive and scalable, the data center network is developing into a foundation across which information, application services and all data center resources, including servers, storage are shared, provisioned, and accessed. Modern data center networks commonly adopt multi-rooted tree topologies. ECMP is often used to achieve high link utilization and improve network throughput. Meanwhile, max-min fairness is widely used to allocate network bandwidth fairly among multiple applications. However, today's data centers usually host diverse applications, which have various priorities (e.g., mission critical applications) and service level agreements (e.g., high throughput). It is unclear how to adopt ECMP forwarding and max-min fairness in the presence of such requirements. We propose Prioritized Max-Min Fair Multi-

 ISBN: 978-1-61208-318-6



(a) (b) (c)

Figure 6: Throughput with high priority for (a) long flows (b)short flows (c)mixed flows

ple Path forwarding (PMP) to tackle this challenge. PMP can prioritize current demands and allocate available bandwidth accordingly. Our performance evaluation results show that PMP can improve application throughput 10-12% on average and increase overall link utilization especially when the total demanded bandwidth close or even exceed the bisectional bandwidth of a data center network.

REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, A Scalable, Commodity Data Center Network Architecture. In SIGCOMM, 2008

[2] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, PortLand: A scalable fault-tolerant layer 2 data center network fabric. In SIGCOMM, 2009

[3] R. Perlman, Rbridges: Transparent routing. In INFOCOMM, 2004.

[4] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, VL2: A Scalable And Flexible Data Center Network. In Proc. of ACM SIGCOMM, 2009.

[5] V. Liu, D. Halperin, A. Krishnamurthy, and T. Anderson, F10: A Fault-Tolerant Engineered Network. In NSDI, 2013

[6] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, HyperX: Topology, Routing, and Packaging of Efficient Large-Scale Networks. In Proc. of SC, 2009.

[7] J. Kim, W. J. Dally, and D. Abts, Flattened butterfly: A Cost-efficient Topology for High-radix networks. ISCA, 2007.

[8] J. Kim, W. J. Dally, S. Scott, and D. Abts, Technology-Driven, Highly-Scalable Dragonfly Topology. In Proc. of ISCA, 2008

[9] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, Jellyfish: Networking Data Centers Randomly. In NSDI, 2012.

[10] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, Hedera: Dynamic Flow Scheduling for Data Center Networks. In NSDI, 2010.

[11] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, Data center TCP (DCTCP). In Proc. of ACM SIGCOMM, 2010.

[12] T. Benson, A. Akella, and D. A. Maltz, Network Traffic Characteristics of Data Centers in the Wild. IMC, 2010.

[13] T. Benson, A. Anand, A. Akella, and M. Zhang, MicroTE: Fine Grained Traffic Engineering for Data Centers. In Proc. of ACM CoNEXT, 2011.

[14] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, DevoFlow: Scaling Flow Management for High-Performance Networks. In Proc. of ACM SIGCOMM, 2011.

[15] OpenFlow Switch Specification (Version 1.1), http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf. (retrieved: Sept. 2013)

[16] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, The Nature of Data Center Traffic: Measurements and Analysis. In IMC, 2009.

[17] T. Benson, A. Akella, and D. A. Maltz, Network Traffic Characteristics of Data Centers in the Wild. In IMC, 2010.

[18] G. Wang, D. G. Andersen, M. Kaminsky, M. Kozuch, T. S. E. Ng, K. Papagiannaki, M. Glick, and L. Mummert, Your data center is a router: The case for reconfigurable optical circuit switched paths. In Hotnets, 2009.

[19] S. Radhakrishnan, M. Tewari, R. Kapoor, G. Porter, and A. Vahdat, Dahu: Commodity Switches for Direct Connect Data Center Networks. In Proceedings of the 9th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS13), October 2013

[20] D. Zats, T. Das, P. Mohan, D. Borthakur, and R. Katz, DeTail: Reducing the Flow Completion Time Tail in Datacenter Networks. In SIGCOMM, 2012.

[21] S. Kandula, D. Katabi, B. Davie, and A. Charny, Walking the Tightrope: Responsive Yet Stable Traffic Engineering. In SIGCOMM, 2005.

[22] S. Fischer, N. Kammenhuber, and A. Feldmann, REPLEX: Dynamic Traffic Engineering Based on Wardrop Routing Policies. In CoNEXT, 2006.

[23] A. Ghodsi, M. Zaharia, S. Shenker and I. Stoica, Choosy: Max-Min Fair Sharing for Datacenter Jobs with Constraints, EuroSys 2013.

[24] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, I. Stoica, and S. Shenker, Dominant resource fairness: Fair allocation of multiple resource types. In NSDI, 2011

[25] Hadoop Capacity Scheduler. hadoop.apache.org/common/docs/r0.20.2/capacity_scheduler.html. (retrieved: Sept. 2013)

[26] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling. In EuroSys 10, 2010.

[27] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, Sharing the data center network. In NSDI, pages 2323, 2011.

[28] M. Casado, D. Erickson, I. A. Ganichev, R. Griffith, B. Heller, N. Mckeown, D. Moon, T. Koponen, S. Shenker, and K. Zarifis, Ripcord: A modular platform for data center networking. UC, Berkeley, Tech. Rep. UCB/EECS-2010-93

 ISBN: 978-1-61208-318-6

PonderFlow: A Policy Specification Language for Openflow Networks

Bruno Lopes Alcantara Batista
Universidade Estadual do Ceará (UECE)
bruno@larces.uece.br

Marcial P Fernandez
Universidade Estadual do Ceará (UECE)
marcial@larces.uece.br

***Abstract*—The OpenFlow architecture is a proposal from the Clean Slate initiative to define a new Internet architecture where network devices are simple, and the control plane and management are performed on a centralized controller, called Openflow controller. Each Openflow controller provides an Application Programming Interface (API) that allows a researcher or a network administrator to define the desired treatment to each flow inside controller. However, each Openflow controller has its own standard API, requiring users to define the behavior of each flow in a programming or scripting language. It also makes difficult for the migration from one controller to another one, due to the different APIs. This paper proposes the PonderFlow, an extension of Ponder language to OpenFlow network policy specification. The PonderFlow extends the original Ponder specification language allowing to define an Openflow flow rule abstractly, independent of Openflow controller used. Some examples of OpenFlow policy will be evaluated showing its syntax and the grammar validation.**

Keywords-Openflow; OpenFlow Controller; Policy-based Network Management; Policy Definition Language

I. INTRODUCTION

The OpenFlow architecture is a proposal of the Clean Slate initiative to define an open protocol that sets up forward tables in switches [1]. It is the basis of the Software Defined Network (SDN) architecture, where the network can be modified dynamically by the user, and the control-plane is decoupled from the data-plane. The OpenFlow proposal tries to use the most basic abstraction layer of the switch to achieve better performance. The OpenFlow protocol can set a condition-action tuple on switches like forward, filter and also, count packets from a specific flow that match a condition.

The network management is performed by the OpenFlow Controller maintaining the switches simple, only with the packet forwarding function. This architecture provides several benefits: (1) OpenFlow controller can manage all flow decisions reducing the switch complexity; (2) A central controller can see all networks and flows, giving global and optimal management of network provisioning; (3) OpenFlow switches are relatively simple and reliable, since forward decisions are defined by a controller, rather than by a switch firmware. However, as the number of switches increases in a computer network and it becomes more complex to manage the switches flows, it is necessary to use a tool to help the network administrator to manage the flows in order to dynamically modify the system behavior.

A policy-based tool can reduce the complexity inherent to this kind of problem, providing a simple way to manage a large network environment, where the behavior of the network assets may change over time. Policy-Based Network Manager (PBNM) is the technology that provides the tools for automated management of networks using policies to abstract the behavior on the environment. The PBNM can help network administrators to manage OpenFlow networks simply defining policies, where a policy is a set of rules that govern the behavior of the system.

This paper presents the PonderFlow, an extension of Ponder policy specification language. Ponder is a declarative, object-oriented language for specifying management and security policy proposed by Damianou et al. [2]. The PonderFlow provides the necessary resources to define or remove flows, grant privileges to a user, add or remove flows (authorization policy) and force a user or system to execute an action before a particular event (obligation policy).

The rest of the paper is structured as follows. In Section II, we present some related work about OpenFlow policy specification languages. Section III introduces the OpenFlow, the Policy-Based Openflow Network Manager (PBONM) architecture and introduces the Ponder specification language. In Section IV, we present the PonderFlow language, and its respective grammar and validation. In Section V, we conclude the paper and present some future works.

II. RELATED WORK

Foster et al. [3] designed and implemented the Frenetic, a set of Python's libraries for network programming that provides several high-level features for OpenFlow/NOX [4] programming issues. Frenetic is based on functional reactive programming, a paradigm in which programs manipulate streams of values, delivering the need to write event-driven programs leading a unified architecture where programs "see every packet" rather than processing traffic indirectly by manipulating switch-level rules. However, the network administrator needs to use a programming language, Python [5] in this case, to define the behavior of OpenFlow network.

Mattos et al. [6] propose an OpenFlow Management Infrastructure (OMNI) for controlling and managing OpenFlow networks and also for allowing the development of autonomous applications for these networks. OMNI provides a web interface with set of tools to manage and control the network, and the network administrators interact through this interface. The outputs of all OMNI applications are eXtensible Markup

 ISBN: 978-1-61208-318-6

Language (XML), simplifying the data interpretation by other applications, agents or human operators. However, the network administrator needs to use a programming language to call any OMNI function using a web Application Programming Interface (API) or access the web interface and proceed manually.

Voellmy et al. proposed Procera [7], a controller architecture and high-level network control language that allow to express policies in the OpenFlow controllers. Procera applies the principles of functional reactive programming to provide an expressive, declarative and extensible language. Users can extend the language by adding new constructors.

The PonderFlow has similarities with Procera and Frenetic, but our main goal is to create a policy specification language decoupled from the conventional programming languages, and also, regardless of the OpenFlow controller used. The PonderFlow language is an extension of Ponder language and can be easily ported to another OpenFlow controller. As Ponder is a well-known policy language, the validations is not necessary. In this work, we used the Java language to implement the parser and lexical analyses in Floodlight OpenFlow controller [8]. This way, we want to achieve a level of independence from the programming language and of the OpenFlow controllers. This paper presents the PonderFlow, an extensible, declarative language for policy's definition in an OpenFlow network.

III. OpenFlow Policy Architecture

The OpenFlow architecture has several components: the OpenFlow controller, one or many OpenFlow devices (switch), and the OpenFlow protocol. This approach considers a centralized controller that configures all devices. Devices should be kept simple in order to reach better forward performance and leave the network control to the controller.

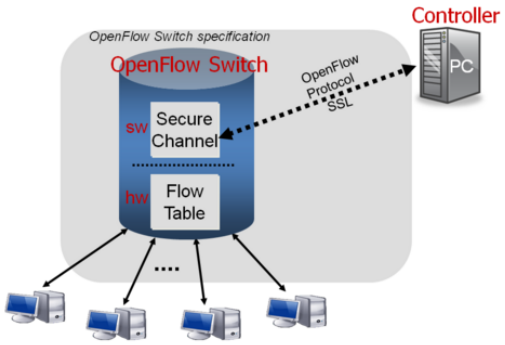


Figure 1. The OpenFlow architecture [1]

The OpenFlow Controller is the centralized controller of an OpenFlow network. It sets up all OpenFlow devices, maintains topology information, and monitors the overall status of entire network. The device is any capable OpenFlow device on a network such as a switch, router or access point. Each device maintains a Flow Table that indicates the processing applied to any packet of a certain flow. There are several OpenFlow controllers available, e.g., FloodLight [8], NOX [4], POX [9], and Trema [10].

The OpenFlow Protocol works as an interface between the controller and the OpenFlow devices setting up the Flow Table. The protocol should use a secure channel based on Transport Layer Security (TLS). The controller updates the *Flow Table* by adding and removing Flow Entries using the OpenFlow Protocol. The Flow Table is a database that contains Flow Entries associated with actions to command the switch to apply some actions on a certain flow. Some possible actions are: forward, drop, and encapsulate.

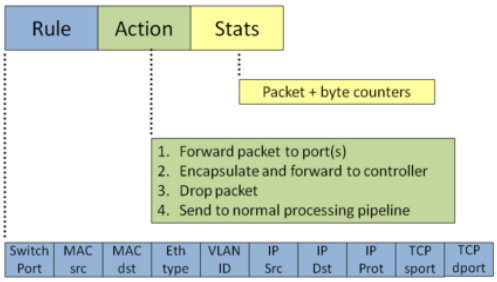


Figure 2. The OpenFlow Flow Entry [11]

Each device has a Flow Table with flow entries as shown in Figure 2. A Flow Entry has three parts: rule match fields, an action and statistics fields and byte counters. The *rule match fields* is used to define the match condition to a specific flow. *action* defines the action to be applied to an exact flow, and *statistical fields* are used to count the rule occurrence for management purposes.

When a packet arrives to the OpenFlow Switch, it is matched against *flow entries* in the *flow table*, and the action will be triggered if the header field is matched and then update the counter. If the packet does not match any entry in the *flow table*, the packet will be sent to the controller over a secure channel. Packets are matched against all *flow entries* based on a prioritization, where each *flow entry* on *flow table* has a priority associated. Higher numbers have higher priorities.

A. Policy-Based Openflow Network Manager

The behavior of an OpenFlow network is defined by flow table entries of the devices (e.g., switch) comprising the network. These entries determine the action to be taken by the device, which may authorize the entry of a package in the device so that it can be forwarded to another device or host or deny the packet in the device. However, some questions arise naturally about: (1) How to create or manage OpenFlow network with controllers currently present? (2) How to delegate or revoke network permissions to a particular user? (3) How to manage the switches flows as the number of hosts and switches increases?

Policy-Based Network Manager (PBNM) has emerged as a promising paradigm for network operation and management, and has the advantage to dynamically change the behavior of a managed system according to the context requirements without the need to modify the implementation of managed system [12]. The general PBNM can be considered an adaptation of the Internet Engineering Task Force (IETF) policy framework to apply to the area of network provisioning and configuration.

With PBNM the management network process can be simplified through of centralization and business-logic abstractions [12]. Centralization refers to the process of configuring all

 ISBN: 978-1-61208-318-6

devices in a single-point (Policy Management Tool (PMT)) instead of reconfiguring the device individually.

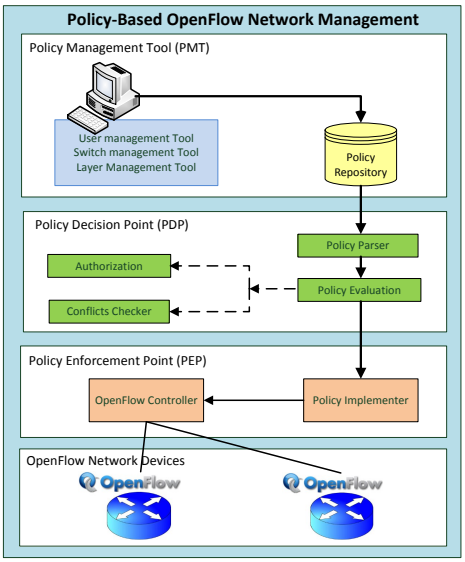


Figure 3. The Policy-Based OpenFlow Network Manager architecture

In a previous work [13], we propose to use the PBNM concepts in OpenFlow networks. PBONM was proposed, a framework based on the IETF policy framework. Ponder language was chosen as the standard policy specification language in the PBONM. The PBONM is depicted on Figure 3. The architecture is divided in the following layers:

Policy Management Tool (PMT): it is a software layer that manages the network users, switches and OpenFlow layers providing the User Interface to enable these features. The Ponder is used to specify the policies in this layer. The Policy Repository (database) will store the policies and other information about of the network.

Policy Decision Point (PDP): it is responsible to interpreting the policies stored in the repository, checks the users' authorization (if the user has permission to add or remove a flow in specific switch), check policy conflicts on database and release the policies to Policy Enforcement Point.

Policy Enforcement Point (PEP): it is responsible to execute the configuration of OpenFlow controller. When the policies are interpreted, OpenFlow flows are generated and forwarded to the OpenFlow controller. So, the OpenFlow controller can enforce these flows on the network.

OpenFlow Network Devices: they are OpenFlow switches controlled by an OpenFlow controller and configured by PEP.

Thus, the network administrator can specify network flows and the users' permission through of a graphical tool using a policy specification language. These policies will be translated to OpenFlow controller API calls and will be applied to the network devices.

B. Ponder: Policy Specification Language

Ponder is a declarative, object-oriented language for specifying security and management policy for distributed object systems proposed by Damianou et al. [2]. The language is flexible, expressive and extensible to cover the wide range of requirements implied by the current distributed systems requirement and allows for the specification of security policies (role-based access control) and management policies (management obligations) [14].

There are four building blocks supported on Ponder, which are: (1) *authorizations*: what activities the subject can perform on the set of target objects; (2) *obligations*: what activities a manager or agent must perform on target objects; (3) *refrains*: what actions a subject must not execute on target objects; (4) *delegation*: granting privileges to grantees.

However, the Ponder language does not support the network flows abstraction. In contrast, OpenFlow architecture works over the network flows concept. To use Ponder in PBONM, an extension to the language is needed, to support the requirement inherent in the new environment. Thus, a network administrator can define flows in a network switch OpenFlow clearly and concisely.

The advantage of using a policy language is that the network administrator only needs to think in an abstract form, how the OpenFlow network should work, without worrying about the implementation details of a specific controller. Unlike other flow language's definition, that requires the administrator to use a programming language [3], [6], [7].

Ponder2 is a re-design of Ponder language and toolkit, maintaining the concepts and the basic constructs [15]. In contrast to the original Ponder, which was designed for general network and systems management, Ponder2 was designed as an extensible framework that can be used to configure more complex services. It uses the PonderTalk, a high-level configuration and control language, and it permits user-extensible Java objects. In our proposal, we prefer to use the original Ponder language because the new functionality of Ponder2 is not necessary. We believe that the concise description of Ponder is easier for a network administrator, unlike the more extensible and complex PonderTalk description.

IV. PONDERFLOW: OPENFLOW POLICY SPECIFICATION LANGUAGE

Ponder is the policy language used to manage security policies and access control. However, the Ponder language is too vague to cover all types of manageable environments [16]. PonderFlow is a policy definition language for OpenFlow networks where your main objective is to specify flows transparently, independent of OpenFlow controller used in the network. The PonderFlow extends the Ponder language [2] to suit the flow definition paradigm of OpenFlow environment.

Some of the Ponder's building blocks were kept and others were not used in favor of simplicity. Nevertheless, even keeping some building blocks from the original Ponder language; the philosophy behind these blocks was changed to suit the paradigm of OpenFlow networks. Furthermore, it was added a way to specify flows through policies, making PonderFlow a declarative scripting language. In this way, the new keyword **flow** is included to specify the flow's characteristics. In the following subsections, the building blocks will be explained, and we will show some examples to manage network flows.

 ISBN: 978-1-61208-318-6

ANTLR framework [17] was used to generate the lexical analyzer and parser grammar in the Java programming language, as well as to generate the images of Abstract Syntax Tree (AST) tree of the building blocks defined in the PonderFlow.

A. Authorization Policies

The authorization policies define what the members within a group (subject) may or may not do in the target objects. Essentially, these policies define the level of access that users possess to use an OpenFlow switches network.

A positive authorization policy defines the actions that subjects are permitted to do on target objects. A negative authorization policy specifies the actions that subjects are not allowed to do on target objects.

This building block is very similar to the original language Ponder, but the focus of this building block in PonderFlow context is in the access by the users in the switches that comprise the OpenFlow network and OpenFlow controller itself.

Listing 1. PonderFlow Authorization Policy Sintax

```
1 inst ( auth+ | auth- ) policyName {
    subject [<type_def>]  domain-scope-expression;
3   target  [<type_def>]  domain-scope-expression;
    [ flow  [<type_def>]  flow-expression; ]
5   action          action-list;
    [ when constraint-expression |
      constraint-flow-expression ];
7 }
```

The syntax of an authorization policy is shown in Listing 1. Everything in bold is language keywords. Choices are enclosed within round brackets **( )** separated by **|**. Names and variables are represented within **< >**. Optional elements are specified with square brackets **[ ]**. The policy body is specified between braces **{ }**.

Constraints are optional in authorization policies and can be specified to limit applicability of policies based on time or attributes values to the objects on which the policy refers.

The elements of an authorization policy can be specified in any order, and the policy name must begin with a letter and contain letters, numbers and underscore in the rest of your name.

The specification of the subject and target may be optionally specified using an Uniform Resource Identifier (URI) that represent the domain of the subject or of the target. Moreover, we can specify the subject type or the target type in the policy definition.

Listing 2. Positive authorization policy example

```
1 inst auth+ switchPolicyOps {
    subject <User>     /NetworkAdmin;
3   target  <OFSwitch>  /Nregion/switches;
    action addFlow(), removeFlow(), enable(), disable();
5 }
```

The Listing 2 shows an example of a positive authorization policy that allows all network administrators to perform the actions of adding flows, remove flows, enable and disable all switches in Nregion. Note that this policy is applied to any flow, and it is similar to conventional Ponder authorization policy. In Figure 4, we show the AST tree of a positive authorization policy from Listing 2.

The language also provides the ability to define policy types, enabling the reuse of policies by passing formal parameters in its definition. Several instances of the same type can be created and adapted to the identical environment through real values as arguments.

Listing 3. Type definition policy sintax

```
1 type ( auth+ | auth- ) policyType ( formalParameters) {
    authorization-policy-parts
3 }
  inst ( auth+ | auth- ) policyName = policyType(
       actualParameters )
```

The authorization policy switchPolicyOps (from Listing 2) can be specified as a type of the subject and target given as parameters as shown in Listing 4.

Listing 4. Type policy definition example

```
  type auth+ PolOpsT(subject s, target <OFSwitch> t) {
2     action load(), remove(), enable(), disable() ;
  }
4 inst auth+ admPolyOps=PolOpsT(/NetworkAdmins,
       /NregionA/switches);
  inst auth+ rsrPolOps=PolOpsT( /Researchers,
       /NregionB/switches);
```

Furthermore, we can use the PonderFlow Authorization Policies to define a flow in the OpenFlow network. A flow is an OpenFlow network path between hosts, independent of the switch quantity. Thus, network administrator does not need to use a programming language like Java, Python or C++, to directly manipulate the OpenFlow network behavior through of the OpenFlow controller.

Listing 5. Type policy definition example

```
1 flow-expression = on = <DPID> ,
          | src = <DPID>/<switch_port> ,
3         | src = <IP-ADDRESS> ,
          | src = <MAC_ADDRESS> ,
5         | dst = <DPID>/<switch_port> ,
          | dst = <IP-ADDRESS> ,
7         | dst = <MAC_ADDRESS> ,
          | by = <DPID> ;
```

To define a flow, we need to use the keyword **flow** in the authorization policy statement. With this keyword, we can define the characteristic of the flow. Furthermore, it is possible define a path restriction where the network administrator can define where the flow must pass.

The Listing 5 shows the grammar of *flow-expression*, where: *DPID* is the switch identification, *src* and *dst* are respectively the source device and destination device, *switch_port* is the incoming packet switch port, *IP-ADDRESS* is a valid IP address and *MAC-ADDRESS* is a valid MAC address.

Listing 6. A PonderFlow authorization policy

```
  inst auth+ flow01{
2   subject <User>    /Users/Students/John;
    target  <Switch> /Uece/Macc/Larces/Switches;
4   flow <Flow> src=00:00:00:2C:AB:7C:07:2A/2 ,
                dst=00:00:00:47:5B:DD:3F:1B/5 ,
6               by =00:00:00:C5:FF:21:7F:3B ,
                    00:00:00:33:45:AF:1C:8A ;
8   action setFlow();
    when src-ip=192.168.0.21,
10       dst-ip=192.168.0.57,
         dst-port=80;
12 }
```

 ISBN: 978-1-61208-318-6

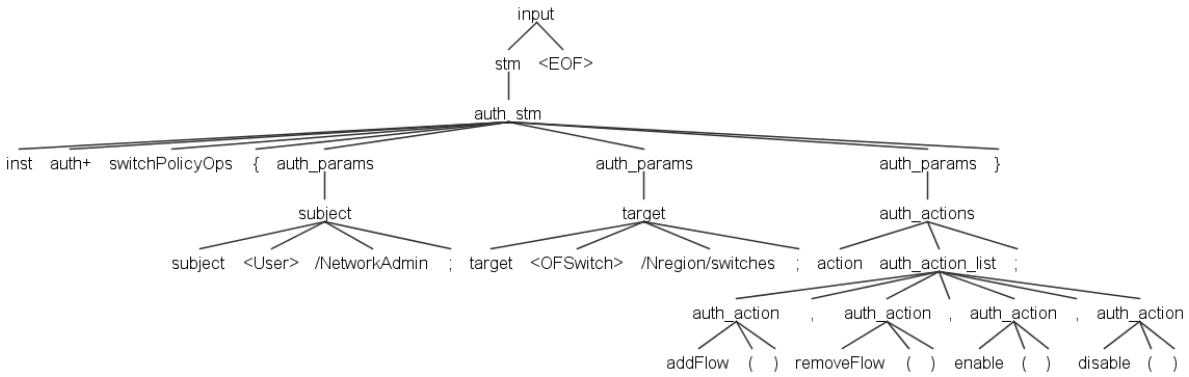


Figure 4. The AST tree for Listing 2 example

TABLE I. OPENFLOW POLICY WILDCARDS

ingress-port	The switch port on which the packet is received
src-mac	The source mac address value
dst-mac	The destination mac address value
vlan-id	The VLAN identification value
vlan-priority	The VLAN priority value
ether-type	The ethernet type value
tos-bits	The ToS bits value
protocol	The IP protocol number used in the protocol field
src-ip	The source IP address value
dst-ip	The destination IP address value
src-port	The source protocol port value
dst-port	The destination protocol port value

TABLE II. OPENFLOW ACTION FIELD

setFlow()	Set the flow(s) in a specified path
delFlow()	Delete the flow(s) in a specified path
setSrcIp(ip-address)	Set the source IP address of the packet
setDstIp(ip-address)	Set the destination IP address of the packet
setSrcMac(mac-address)	Set the source MAC address of the packet
setDstMac(mac-address)	Set the destination MAC address of the packet
setSrcPort(port)	Set the source port of the packet
setDstPort(port)	Set the destination port of the packet
setVlanId(integer)	Set the VLAN of the packet
setVlanPriority(integer)	Set the VLAN priority of the packet

The example in Listing 6 authorizes a flow to user /User/Students/John (**subject**), on the switches of domain */Uece/Macc/Larces/Switches*, set flows (**action**) on the network to establish a path starting from the switch with Datapath ID (DPID) 00:00:00:2C:AB:7C:07:2A on the port 2 (**src**) and ending in the switch with DPID 00:00:00:47:5B:DD:3F:1B on port 5 (**dst**), passing by the switches with DPID 00:00:00:C5:FF:21:7F:3B and 00:00:00:33:45:AF:1C:8A (**by**) when the source IP address of the flow is 192.168.0.21, the destination IP address 192.168.0.57 and the protocol destination port is 80.

PonderFlow specifies a set of default actions for flow definition, but the developers are free to add more actions to the language. The default actions are listed in Table II. The Listing 7 defines a policy which user Alice can set a flow action that changes the source IP address of the packet to 10.23.45.65 when the destination IP address is 10.23.45.123 on the switch with DPID 00:00:00:4F:32:1D:56:9C.

Listing 7. The flow definition that change the source ip address

```
inst auth+ flow02{
    subject <User>    Alice;
    target  <Switch> 00:00:00:4F:32:1D:56:9C;
    action setSrcIP('10.23.45.65');
    when dst-ip=10.23.45.123;
}
```

Furthermore, it is possible to define a policy to be applied in a specific switch and not a path. This is desirable when the network administrator wishes to add or remove a particular flow in a specific switch, in this way, the network administrator changes the network behavior in a single point on the network.

B. Obligation Policies

Obligation policies allow to specify actions to be performed by the network administrator or by the OpenFlow controller when certain events occur in an OpenFlow network and provide the ability to respond any change in circumstances.

These policies are event-triggered and define the activities subjects (network administrator or OpenFlow controller) must perform on objects within the target domain. Events can be simple, e.g., an internal timer, or more complex, starting by reading some kind of sensor, e.g., a network card stopped.

This building block is very similar to the original language Ponder, but in the context of PonderFlow, including flow definition. This block sets an obligation for the network administrator or the OpenFlow controller performs some action, or simply is notified, when a particular event occurs.

Listing 8. Obligation policy sintax

```
inst oblig policyName {
    on event-specification ;
    subject [<type_def>] domain-Scope-Expression ;
    [ target [<type_def>] domain-Scope-Expression ; ]
    do obligation-action-list ;
    [ catch exception-specification ; ]
    [ when constraint-Expression ; ]
}
```

The syntax of obligation policies is shown in Listing 8. The required event specification follows the **on** keyword. The target element is optional in obligation policies. The optional catch-clause specifies an exception that is performed if the actions fail to execute, for some reason.

In Listing 9, the obligation policy is triggered when a failure on adding a flow occurs. The network administrator

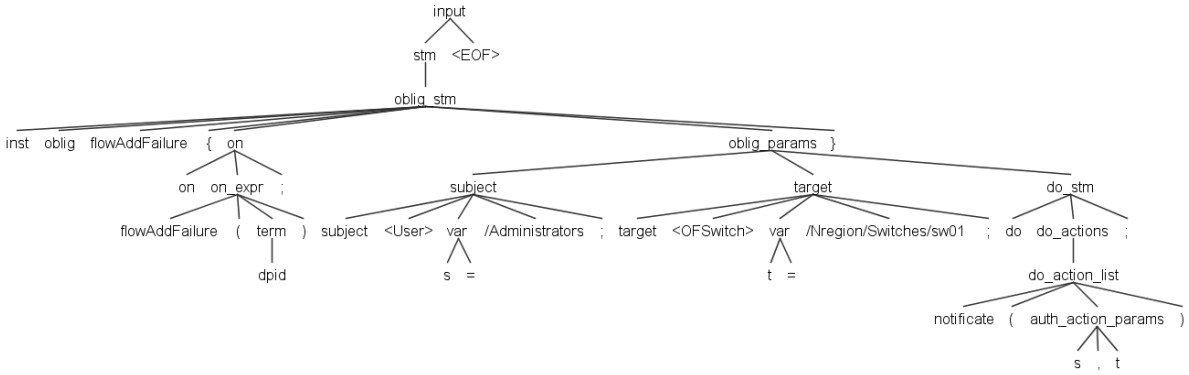


Figure 5. The AST tree for Listing 9 example

will be notified when this event occurs, and he will receive the switch ID where it happened. Figure 5 shows the AST tree of Listing 9.

Listing 9. Obligation policy sintax

```
inst oblig flowAddFailure {
  on flowAddFailure(dpid) ;
  subject <User> s=/Administrators ;
  target <OFSwitch> t = /Nregion/Switches/sw01 ;
  do notificate(s, t) ;
}
```

To perform an obligation policy, it is required that the user has an authorization over the target. This can be specified with an authorization policy. If there is no authorization policy specifying who can perform a particular action, the obligation policy will produce an exception error (depends on the implementation), and the policy will not be applied in the system.

V. Conclusion and Future Works

The paper described the PonderFlow language, a new policy specification language for OpenFlow networks. With this language, the network administrator does not need to know a programming language, like Java, Python or C++, to specify the policy of an OpenFlow network. The language building blocks are simple and concise to define flow policy. The PonderFlow grammar was presented as well as some examples of usage and their AST tree representation. The grammar was tested using the ANTLR framework, which generates the parser and the lexical analyser for the Java programming language.

As a future work, we will extend the Ponder language to use the OpenFlow 1.3 specification. This work used the OpenFlow 1.0 specification for the PonderFlow because most of the commercial switches support only this version. Another point that should be studied is the treatment of policy's conflicts, where a network administrator can, by accident or malpractice, declare two or more conflicting policies. It is necessary to perform an assessment on all policies before applying them on OpenFlow controller.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, 2008, pp. 69–74.

[2] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," in Proceedings of the International Workshop on Policies for Distributed Systems and Networks (POLICY '01). London, UK, UK: Springer-Verlag, 2001, pp. 18–38. [Online]. Available: http://dl.acm.org/citation.cfm?id=646962.712108

[3] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker, "Frenetic: a network programming language," SIGPLAN Not., vol. 46, no. 9, Sep. 2011, pp. 279–291. [Online]. Available: http://doi.acm.org/10.1145/2034574.2034812

[4] NOXRepo.org, "NOX Openflow Controller," Last accessed, Aug. 2013. [Online]. Available: http://www.noxrepo.org/nox/about-nox/

[5] G. VanRossum and F. L. Drake, The Python Language Reference. Python Software Foundation, 2010.

[6] D. M. F. Mattos, N. C. Fern, V. T. D. Costa, L. P. Cardoso, M. Elias, M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "Omni: Openflow management infrastructure," Paris, France, 2011.

[7] A. Voellmy, H. Kim, and N. Feamster, "Procera: a language for high-level reactive network control," in Proceedings of the first workshop on Hot topics in software defined networks, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 43–48. [Online]. Available: http://doi.acm.org/10.1145/2342441.2342451

[8] D. Erickson, "Floodlight Java based OpenFlow Controller," Last accessed, Aug. 2013. [Online]. Available: http://floodlight.openflowhub.org/

[9] NOXRepo.org, "POX Openflow Controller," Last accessed, Aug. 2013. [Online]. Available: http://www.noxrepo.org/pox/about-pox/

[10] NEC Corporation, "Trema Openflow Controller," Last accessed, Aug. 2012. [Online]. Available: http://trema.github.com/trema/

[11] B. Heller, "Openflow switch specification, version 1.0.0," Dec. 2009. [Online]. Available: www.openflowswitch.org/documents/openflow-spec-v1.0.0.pdf

[12] D. C. Verma, "Simplify network administration using policy-based management," IEEE Network, March/April 2002.

[13] B. L. A. Batista, G. A. L. de Campos, and M. P. Fernandez, "A proposal of policy based OpenFlow network management," in 20th International Conference on Telecommunications (ICT 2013), Casablanca, Morocco, May 2013.

[14] M. Sloman, "Policy driven management for distributed systems," Journal of Network and Systems Management, vol. Vol.2, no. No 4, 1994.

[15] K. Twidle, E. Lupu, N. Dulay, and M. Sloman, "Ponder2-a policy environment for autonomous pervasive systems," in Policies for Distributed Systems and Networks, 2008. POLICY 2008. IEEE Workshop on. IEEE, 2008, pp. 245–246.

[16] T. Phan, J. Han, J.-G. Schneider, T. Ebringer, and T. Rogers, "A survey of policy-based management approaches for service oriented system," 19th Australian Conference on Software Engineering, 2008.

[17] T. Parr, "ANTLR: ANother Tool for Language Recognition," Last accessed, Aug. 2013. [Online]. Available: http://www.antlr.org/

 ISBN: 978-1-61208-318-6

Proposal for a New Generation SDN-Aware Pub/Sub Environment

Toyokazu Akiyama, Yukiko Kawai
Kyoto Sangyo University
Kyoto, Japan
{akiyama, kawai}@cc.kyoto-su.ac.jp

Katsuyoshi Iida
Tokyo Institute of University
Tokyo, Japan
iida@gsic.titech.ac.jp

Jianwei Zhang, Yuhki Shiraishi
Tsukuba University of Technology
Ibaraki, Japan
{zhangjw, yuhkis}@a.tsukuba-tech.ac.jp

***Abstract*—Software defined networks are now attracting attention from network engineers due to their flexible controllability. However, the ways in which they interact with applications, specifically, their deployment image, remains unclear. In order to investigate possible application interactions with a software defined network, attention is focused on target applications that depend on a publish and subscribe environment, with the goal of proposing a new environment that will coordinate application layer requirements and network layer services. However, since our project is still in the start-up phase, this paper will discuss the issues that will need to be resolved in order to utilize software defined network functions based on our publish and subscribe environment proposal, and our approaches to resolving them.**

Keywords-Software defined network (SDN); Publish/subscribe communication model; Overlay network; P2P middleware

I. INTRODUCTION

Recently, a number of Software Defined Network (SDN) products such as OpenFlow have been released, and work has progressed towards deploying them into productive networks [1]. SDNs provide flexible network control functions from application programs. However, it is not yet clear what kind of cross-layer control is most suitable for SDN use. Therefore, we have investigated this topic by focusing on target applications.

In this paper, the communication models used by recent Internet applications, which include Social Networking Services (SNSs) [2], groupware, streaming video, and others, will be considered. Most such applications are based on the publish and subscribe (pub/sub) communication model [3][4], which requires asynchronous message transfer and multi-source message distribution functions. Currently, since such pub/sub environments are normally only constructed as backend services for Web applications, they can only be used in data center networks. However, it is believed that if an open pub/sub environment could be provided for end-users, it could enter usage as a general-purpose middleware that provides coordination between the application and network layers. Accordingly, this study focuses on the development of an open pub/sub environment that is tightly coupled with an SDN.

Before a pub/sub environment can be directly utilized from the end-user applications, the relationship between the end user and the environment must be abstracted. In this paper, the concept "topic" is used to abstract user interests and message distribution in a pub/sub network. Once such an abstraction is provided, determining how to extract topics from user behaviors and considering the best way to optimize the resulting topic-based pub/sub network become the main targets of discussion.

However, since topic extraction alone is insufficient to control burst traffic, estimations must be performed to properly handle their impact. Here, if a drastic "topic" change can be identified, it will be referred to as a "cyberspace event". Additionally, if changes to network traffic can be traced, they will be referred to as "network events". Finally, if the "cyberspace events" are found to have correlations with the "network events", and, if future "cyberspace events" can be estimated, they can be used in advance for traffic engineering purposes.

One goal of our research is the construction of an advanced pub/sub infrastructure that weaves the user's behavior and the pub/sub network together by abstracting his or her important and/or changing interests as "topics" and "events", which include not only topic changes but also network traffic changes. Currently, since there is no strict definition for "topics", it is necessary to consider how to define and implement the term. In this paper, the issues related to this effort and our approach to resolving them will be described, in order to work towards constructing an advanced topic-based pub/sub environment that takes into consideration cyberspace and network events.

The remainder of this paper is organized as follows. In Section II, an overview structure of the proposed pub/sub environment is described and issues that must be resolved in order to establish our new pub/sub environment are outlined. Topic-based pub/sub implementation is discussed in Section III. In Section IV, ways to extract topics from user behaviors are discussed, and an approach that can be used to extract cyberspace and network events is examined in Section V. In Section VI, our related work will be shown. Finally, this paper will be concluded in Section VII.

II. PROPOSAL OVERVIEW

Fig. 1 shows an overview structure of the proposed pub/sub environment. To provide a pub/sub environment to end-users, middleware must be used to map application layer requests to network layer services. In our research, P2P Interactive Agent eXtensions (PIAX) [5], which is a peer-to-peer (P2P) middleware implementation, is employed as a pub/sub network frontend. PIAX can provide pub/sub functions via a number of different overlay networks including Skip Graph (SG) and Multi-Key Skip Graph (MKSG) [6]. In SG, any peer with a key can easily reach all the other peers that possess the same key. In MKSG, every

 ISBN: 978-1-61208-318-6

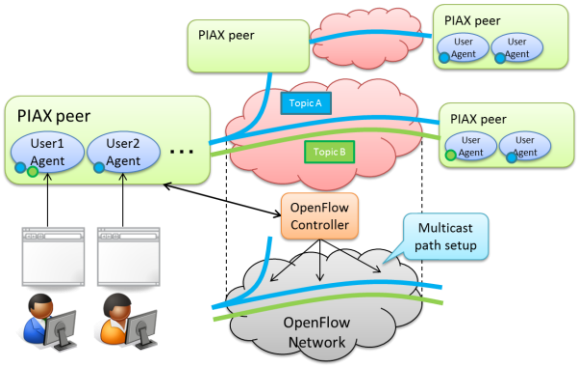


Figure 1. An overview of the proposal.

peer has the same abilities, but they can also possess multiple keys simultaneously. Thus, regarding keys as "topics" provides a straightforward approach for implementing the proposed pub/sub environment, so the base pub/sub functions are covered by using PIAX in our proposal. However, since PIAX is a P2P middleware, it does not have the ability to optimize the lower layer. Several approaches to optimize an overlay network by considering the lower layer information, such as Application Layer Traffic Optimization (ALTO) [7], and several other methods [8][9], have been researched. However, each of those approaches pose particular issues. For example, ALTO requires the addition of a special node to solve topology mismatching. In contrast, the approaches in [8] and [9] do not require special nodes, but both require complicated control and use an application level measurement method that lacks sufficient accuracy. Furthermore, they do not have a direct network equipment control function. In our approach, attempts are made to resolve these issues by integrating an SDN, specifically OpenFlow, into the pub/sub environment. The details are described in Section III.

The client side is assumed to be a Web browser or a mobile application that can access the pub/sub environment via PIAX, as shown in Fig. 1.

In order to bind a user to the user's agent, user authentication and rendezvous peer discovery are required. Here, BrowserID [10], which is an open decentralized protocol for authenticating users based on email addresses, is employed. While BrowserID is based on the low cost Public Key Infrastructure (PKI), as described in [11], if the initial authentication method and certificate duration are properly selected, it provides a good candidate for the authentication layer of our proposal. As for the rendezvous peer discovery, several approaches, such as the ones taken in Content Delivery Network (CDN) services, are available. However, if information related to the user status such as the type of network device that is currently available can be detected, it will enable the client side system to further optimize the route selection process. Thus, more investigation into the "user agent binding" issue needs to be performed, especially as related to the "topic" extraction process, as described below.

Since topic definition is not easy for end-users to perform, the user agent plays an important role in coordinating user requirements and topic-based pub/sub functions. This means that finding the most suitable way of extracting topics from each application and overlaying them into the pub/sub environment becomes a critical problem. Our current plan for resolving this problem is described in Section IV.

Topic extraction enables us to map the end-users into the network. It also gives us the opportunity to analyze any existing correlations between end-user behavior and network traffic characteristics. To investigate their applicability to traffic engineering, it is necessary to extract events from website and traffic data archives, and then to analyze the data in order to identify correlations. This process is described in detail in Section V. After extracting events, they can be utilized to control the pub/sub environment.

III. TOPIC-BASED PUB/SUB IMPLEMENTATION

As described above, topic-based pub/sub is basically provided by PIAX using MKSG and it currently does not consider the lower layer environment. Therefore, to solve topology mismatching, a lower layer information server like ALTO, or a direct control interface between the application layer and the lower layer, must be provided. In our proposal, the latter approach is adopted to minimize backend traffic.

When a user publishes a message via a pub/sub network, PIAX uses an overlay to transfer the message. If the logical path from the publisher to all the subscribers follows the graph shown at the top of Fig. 2, the message is transferred using the process shown in the bottom of the figure. In this case, since PIAX peers do not take their own physical location into consideration when generating a logical link, each message makes a round trip between the two switches. To reduce such wasteful traffic, the following two approaches can be adopted.

A. User and topic migration based on the OpenFlow information

In the case of Fig. 2, if the three users shown can be hosted on the same Personal Computer (PC), backend traffic can be reduced. Therefore, as shown in Fig. 2, (1), agent migration can reduce traffic. However, if the number of users increases, it becomes difficult to host them all on the same node. Furthermore, as described in Section II, it is important to consider rendezvous peer optimization simultaneously because the optimal peer to connect for a user device may be changed by the user agent migration. Thus, when users and topics are relocated properly based on the physical topology, the frontend and backend traffic can be reduced.

 ISBN: 978-1-61208-318-6

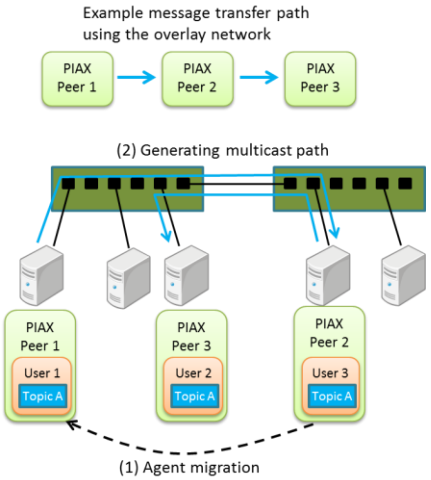


Figure 2. The possibilities to reduce backend traffic in a pub/sub network.

In the case of ALTO, the server provides lower layer information. In the case of OpenFlow, a controller node can also become an ALTO server if the lower layer status grasping function is implemented. For example, OpenFlow switch topology can be detected by using Link Layer Discovery Packets (LLDPs). This, in turn, allows PIAX to relocate user agents based on the topology information provided by the OpenFlow controller. Therefore, determining how to implement an ALTO-like function and utilizing it for user agent relocation are specific issues that need to be resolved.

B. Cross layer optimization

If user agent migration cannot be used for backend traffic reduction because it would exceed the upper limits of the node's capability, further optimization of the lower layer capability is required. For example, if a multicast path can be constructed among topic subscriber peers via OpenFlow, the total traffic will be properly reduced (Fig. 2, (2)). This requires a mapping function between the overlay network and the multicast network. An approach that can be used for constructing a multicast network in an OpenFlow network is described in [12].

IV. TOPIC EXTRACTION FROM USER BEHAVIOR

In this section, the issues related to extracting topics from user behavior are discussed. As described in Section II, the user agent must coordinate user requirements and topic-based pub/sub functions. In the pub/sub environment, entities are categorized by topics, which the user agent and client side application must translate into concepts that users can understand. Here, it is assumed that the client side is a Web browser or a mobile application, and that the Web browser functions can be extended using browser plug-ins.

For example, in our application [13], the system provides Page-Centric Communication (PCC), with which users can communicate with other users over the webpage they are

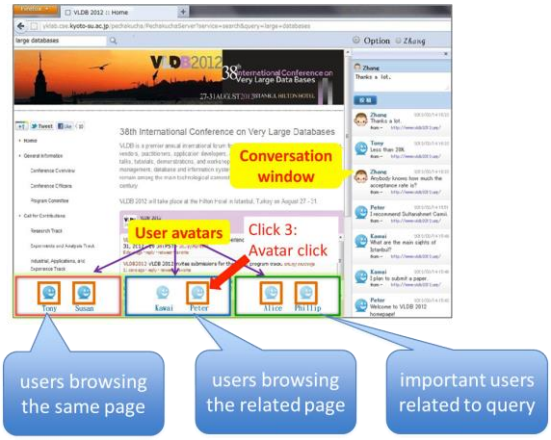


Figure 3. An example of images of page-centric communication.

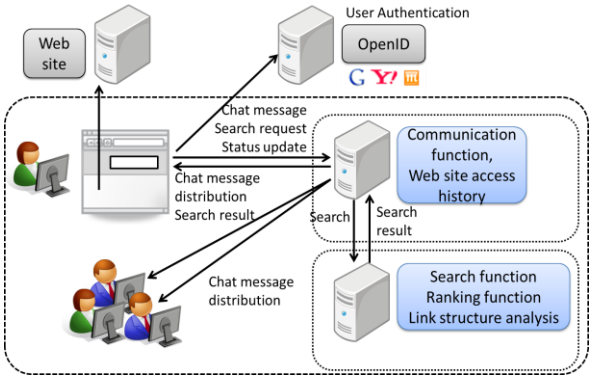


Figure 4. Current system structure of PCC.

visiting (Fig. 3). In such cases, a webpage Uniform Resource Identifier (URI) can be a pub/sub topic. Additionally, in a PCC system, a user can search for both webpages and the other users using the search function provided by the system. Thus, search keywords and webpage visit histories may help extract user interests. Fig. 4 shows the current PCC system structure. Since the current system has scalability problems, the proposed pub/sub environment may provide a solution for that issue.

Web browsers also have interfaces for obtaining user location information. These include, for example, the Geolocation Application Programming Interface (API) [14], which is currently used to attach the user location to the SNS messages. It is also used for extracting user interests. Fig. 5 shows a sample extraction approach. In this example, the correlation between a user's interest and the distance from a target location, which can also be a topic, such as Tokyo Station, is investigated.

Once a topic is extracted from a user's status, it can be mapped to the pub/sub network environment. This enables us to issue automatic topic subscriptions, which might be applicable to local disaster warning services.

Here, in the case of PCC, users can be mapped to the topics related to the webpage URI. However, if the user

 ISBN: 978-1-61208-318-6

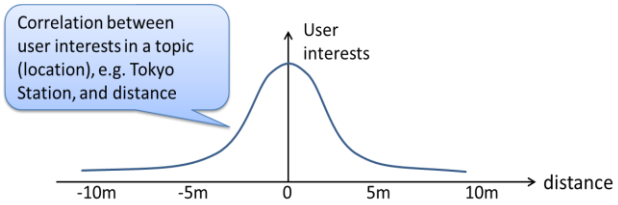


Figure 5. Correlations between user interests and distance to topics.

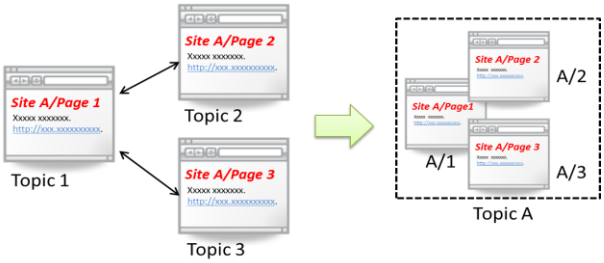


Figure 6. Topic clustering example in the Page Centric Communication System.

frequently moves among webpages, it may not be suitable to choose URIs as topics. In such cases, topic clustering may be required, as shown in Fig. 6.

As described above, topic extraction basically depends on the application specifications. To give an overview of the problem, an attempt to construct a prototype implementation and provide a framework that can be used to simplify the customization of the user agent will be described. In this example, the P2P network property defined in Web Real-Time Communication (WebRTC) [15] may be an appropriate candidate for the interface between a browser and a user agent. Interface standardization of this type would add practicality to our approach.

V. EVENT EXTRACTION

If topics related to the current user's status can be extracted, the migration of user agents and topics for topic-clustering purposes may help normal traffic reduction. However, burst traffic caused by a "cyberspace event", such as SNS messages resulting from a TV program, may exceed the allowable amount of messages per node. If the cyberspace event shows correlation to a severe traffic change (network event), and if the event can be predicted in advance, efforts can be made to prepare for the burst traffic by relocating user agents and constructing a proper multicast path.

To investigate event extraction possibilities, the correlation between the Web archive data and the traffic capture archive will be analyzed. From Web archives, especially news sites, real world and cyberspace events will be extracted, while network events can be extracted from traffic capture archives. If any correlations are identified between them, it indicates the possibility that traffic change events can be estimated in advance because news sites usually include preliminary announcements of newsworthy events.

In the Web space analysis process, the quality and credibility of the contents become important factors. In the analysis of online news archives, a simple search index mining process can be used to find terms representing fresh topics [16]. It is also possible to estimate the focus time of webpages, that is, the time periods to which the content of pages refers. Analysis methods of this type can be used for notable topics and events.

When extracting network traffic change events, existing traffic analysis methods can be used. For example, Hurst exponent analysis results of Measurement and Analysis on the Wide Internet (MAWI) [17] and Cooperative Association for Internet Data Analysis (CAIDA) [18] traffic data are being investigated as methods for detecting network events. Furthermore, it is expected that traffic estimation using time series analysis can be applicable when making minor adjustments to the event period. However, event base estimations of this type are usually difficult to apply to strict traffic control and it would be a challenging theme to implement.

VI. RELATED WORK

As previously mentioned in Section II, our proposal is very similar to ALTO. Thus, while OpenFlow controller is used as a lower layer information collector, it can also function as an ALTO server. The difference is that OpenFlow can also control the lower layer from the application layer.

Bothelho *et al.* [19] proposed the construction of a distributed OpenFlow controller that functions in a way that is very similar to our proposal from the viewpoint of integrating multiple OpenFlow networks. In our proposal, we focus primarily on the consistency and fault-tolerance of the controller while also abstracting the relationship between the application and network layers as well as the introduction of effective interactions between them.

While our proposal is also deeply related to information-centric networking and content-centric networking (ICN/CCN) [20], the most significant difference is that ICN/CCN assumes intermediate nodes have caching ability. In some applications, if the cache ability is required in our environment, it can be established at the user agent or the logically intermediate peer of the overlay network. While this is not as efficient as ICN/CCN, our approach assumes service as an intermediate node just as OpenFlow switches, thus simplifying implementation and minimizing the transfer of multicast path data. Among ICN/CCN projects, the Publish Subscribe Internet Routing Paradigm (PSIRP) [3] is more similar to our proposal. However, when our approach is compared to the PSIRP implementation, which has a high performance lower layer pub/sub environment, it was found that our approach utilizes existing methods more positively, and is simpler than PSIRP as a result. That being said,

 ISBN: 978-1-61208-318-6

usability and performance levels will need to be compared in future work.

VII. CONCLUSION AND FUTURE WORK

In this paper, a new pub/sub environment that can be used with SDNs, especially OpenFlow, was proposed and issues related to using SDN functions in the proposed environment were discussed. In our proposal, the PIAX middleware abstracts the end user's behavior and pub/sub network characteristics as "topics" and "events" to map the application layer requests to network layer services. That enables us to optimize the environment. The possibilities of network optimization in the proposed environment and several approaches that can extract the end user's behavior and network characteristics were also explored. Currently, we are developing a prototype which has a function to optimize multicast communication as discussed in section III. In the future, efforts will continue to develop and investigate the validity of our proposal from various aspects. For example, a performance evaluation of the prototype system by using "topics" and "events" extracted from existing SNSs can be a starting point.

ACKNOWLEDGMENT

This study is partially supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) and the National Institute of Informatics's joint research project. The authors would also like to express their sincere thanks to the MAWI and CAIDA project members for their assistance in this study.

REFERENCES

[1] "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation White paper, http://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf, retrieved on December 2013.

[2] D.M. Boyd, and N.B. Ellison, "Social Network Sites: Definition, History, and Scholarship," Journal of Computer-Mediated Communication, Vol. 13, Issue 1, Dec. 2007, pp.210–230, doi: 10.1111/j.1083-6101.2007.00393.x.

[3] Publish-Subscribe Internet Routing Paradigm (PSIRP), http://www.psirp.org/, retrieved on December 2013.

[4] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: line speed publish/subscribe inter-networking," Proc. of the ACM SIGCOMM 2009 conference on Data communication (SIGCOMM '09). Oct. 2009, pp.195-206, DOI=10.1145/1592568.1592592 http://doi.acm.org/10.1145/1592568.1592592.

[5] Y. Teranishi, "PIAX: Toward a Framework for Sensor Overlay Network," Proc. of 6th Annual IEEE Consumer Communications and Networking Conference, Jan. 2009, pp. 1-5.

[6] Y. Konishi, M. Yoshida, Y. Teranishi, K. Harumoto, and S. Shimojo, "A Proposal of a Multi-key Extension of Skip Graph," IPSJ SIG Notes, Vol. 2007 No. 58, June 2007, pp.25-30.

[7] IETF Application-Layer Traffic Optimization (alto) Working Group, http://datatracker.ietf.org/wg/alto/, retrieved on December 2013.

[8] Y. Liu, X. Liu, L. Xiao, L. Ni, and X. Zhang, "Location-Aware Topology Matching in P2P Systems," Proc. of IEEE INFOCOM, vol. 4, Mar. 2004, pp. 2220-2230.

[9] H. Hsiao, H. Liao, and C. Huang, "Resolving the Topology Mismatch Problem in Unstructured Peer-to-Peer Networks," Parallel and Distributed Systems, IEEE Transactions on, Vol. 20, Issue 11, Nov. 2009, pp. 1668-1681.

[10] Mozilla Persona, https://www.mozilla.org/persona/, retrieved on December 2013.

[11] T. Akiyama, T. Nishimura, K. Yamaji, M. Nakamura, and Y. Okabe, "Design and Implementation of a Functional Extension Framework for Authn & Authz Federation Infrastructure Using Web Browser Add-on," Proc. of 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), Mar. 2013, pp. 389-396.

[12] D. Kotani, K. Suzuki, and H. Shimonishi, "A Design and Implementation of OpenFlow Controller Handling IP Multicast with Fast Tree Switching," Proc. of 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT), Jul. 2012, pp. 60-67.

[13] Y. Shiraishi, J. Zhang, Y. Kawai and T. Akiyama, "Simultaneous Realization of Page-centric Communication and Search," Proc. of ACM CIKM Conference 2012, demo paper, Oct. 2012, pp. 2719-2721.

[14] W3C Geolocation Working Group, http://www.w3.org/2008/geolocation/, retrieved on December 2013.

[15] Web Real-Time Communication, http://www.webrtc.org/, retrieved on December 2013.

[16] A. Jatowt, Y. Kawai, and K. Tanaka, "Calculating content recency based on timestamped and non-timestamped sources for supporting page quality estimation," Proc. of the 2011 ACM Symposium on Applied Computing (SAC'11), Mar. 2011, pp. 1151-1158.

[17] Traffic Archive maintained by MAWI Working Group of WIDE Project, http://mawi.wide.ad.jp/mawi/, retrieved on December 2013.

[18] The Cooperative Association for Internet Data Analysis, http://www.caida.org/, retrieved on December 2013.

[19] F. Botelho, F. Ramos, D. Kreutz, and A. Bessani, "On the feasibility of a consistent and fault-tolerant data store for SDNs," Proc. of the Second European Workshop on Software Defined Networks (EWSDN 2013), Oct. 2013.

[20] G. Xylomenos, et al., "A Survey of Information-Centric Networking Research," Communications Surveys & Tutorials, IEEE, Jul. 2013, pp. 1-26.

 ISBN: 978-1-61208-318-6

Geo-Coded Environment for Integrated Smart Systems

Kirill Krinkin
Open Source and Linux Lab
Academic University RAS
Saint-Petersburg, Russia
kirill.krinkin@fruct.org

Kirill Yudenok
Department of Software Engineering
Saint-Petersburg Electrotechnical University
Saint-Petersburg, Russia
kirill.yudenok@gmail.com

***Abstract*—Smart Systems provide novel enabling functionalities and as such are currently a driving force behind product innovation. Smart Systems are, therefore, crucial for the competitiveness of companies and entire industry sectors. Geo-tagging and smart spaces are two promising directions in modern mobile market. Geo-tagging allows to markup any kind of data by geographical coordinates and time. This is the basis for defining geographical context which can be used in different types of applications e.g., semantic information search, machine-to-machine interactions. Smart spaces as the basis for seamless distributed communication field for software services provides semantic level for data processing. The paper is targeted to discuss opportunity of using geo-coded smart spaces in integrated Smart Systems.**

Keywords-geo-tagging; geo-coding; Smart Spaces; Smart System; LBS.

I. INTRODUCTION

Nowadays, we have two most promising software trends – location based services and pervasive smart environments (smart spaces). Both of them will be a base for user- and machine-oriented proactive services. Smart spaces should provide continuous distributed semantic data and communication field for software services, which is being run on personal devices and autonomous computers and robots. The most desired features of coming software is pro-activeness and context awareness, i.e., services will be able to adapt to the user's needs and situations and be able to manage decisions and behaviors on behalf of the user [1]. One of the important part of context is location-based data. These data are being used for two purposes: for clarifying semantic meaning of queries (when service retrieves the data from smart environment) and for limitation of space of search (usually, there is no point to make global search). Geo-coding (or geo-tagging) is the technique of markup real or virtual object by adding geographical coordinates and time. If we consider software, we have only virtual (or digital) objects like media, events, documents, etc. So far, smart spaces and geo-tagging systems are being developed mostly separately, there are only few works [2][3][4], where software design of smart spaces and geo-tagging integration are discussed.

This paper discusses the definition of a Smart System, based on the creation of an integrated platform as a part of the device for implementing the basic Smart System properties, Smart System use-cases, its architecture and criteria for the analysis of the constructed system.

This document proceeds as follows. Section II provides our definition for Geo Codes Smart System. Section III gives geo-coding problem for smart spaces. System Requirements are discussed in Section IV. In Section V, a high level design is considered. Section VI provides platform integration agent architecture. Smart-M3 and Geo2Tag data integration principles are presented in Section VII and the conclusion is presented in Section VIII.

II. GCSS SMART SYSTEM DEFINITION

A Smart System, called intellectual integrated system, has the following main features:

- System with a clear goal, which determines the directionality of the system;
- System receiving information from the outside world, a person, other systems (data acquisition);
- System responsible for processing information and making decisions to achieve the goals of the system.

Smart Systems must have the following properties:

- *Autonomy* – ability to operate without human intervention or other systems;
- *Openness* – the independent ability to interact with the physical and virtual worlds objects (systems, tools, people), the collection of information and influence to them. It also includes the ability to use and provide external interfaces;
- *Context-awareness* – the ability to independently collect contextual data and analyze the situation;
- *Self-organization* – the ability to maintain the autonomy, to control their own parameters and to select behavior strategies;
- *Purposefulness* – the presence of individual or collective goals and the ability of strategy synthesis and implementation;
- *Pro-activity* – the ability to predict the evolution of the situation in the future (see the decision tree), to determine the parameters of the desired impact and exercise influence.
- *Cooperativeness* – the ability to interact with other systems and/or the person to achieve a goal and assist in achieving the goals of other systems.

There are two possible views on a Smart System: interconnected devices and the device itself. Both of them rely on smart space middleware which provides semantic information sharing facility.

At the moment, there are not effective approaches for markup semantic data by temporal and spatial context for

 ISBN: 978-1-61208-318-6

using in integrated Smart Systems. For instance, if we have presented coordinates and time as traditional Resource Description Framework (RDF) triples, the system performance will be not acceptable, due the big amount of data for processing. On the other hand, there are number of systems for fast temporal and spatial search and filtration. For most of integrated Smart Systems next functions are absent, but required:

- Search objects (usually RDF triples) by given time intervals;
- Defining set of objects which are enclosed inside geographical region or defined spatial structure (buildings, squares, etc.).

The main goal of this work is to suggest an approach for building integrated Smart Systems with using such data model and program interfaces, when advantages of using semantic and geographical markup are available at the same time. In the first instance, integrating Smart Systems are considered.

In other words, we need to design a system, which includes components to perform main and missing Smart System functions. Each system component is responsible for the execution own functions and also provides an Application Programming Interface (API). The integrated system should have a common communication interfaces, protocols and programming interfaces for interaction with other Smart Systems.

As initial system components, we have chosen Smart Spaces [5], Smart-M3 [6] platform, and the Internet of Things LBS Geo2Tag [7] platform, as ones of the fastest growing platforms of these areas. Smart-M3 platform provides a common communication field for cooperation and allows processing and storing semantic information (knowledge). LBS Geo2Tag platform is responsible for the provision of geospatial data from a variety sources.

To build a new Smart System based on selected technologies, we must consider the following aspects:

- Integrated system architecture development;
- Unification of general platforms levels (Smart-M3 and Geo2Tag);
- Common protocols and communication interfaces between device;
- Behavior model.

The main feature of the Smart System discussed in this project is the ability to connect the location data to any object in smart space.

An area such as Smart Systems can be used in various spheres of human activity:

- Space industry;
- Automotive industry;
- Information and telecommunication area;
- Internet of Things;
- Energetic industry;
- Medical area;
- Privacy and security.

In the solutions proposed in this paper, the Smart System allows determining the location of each space object (thing, entity) in time. It can be used for spatial and geographical context clarification in order to increase context awareness of user-oriented services.

As one can be seen from the further evolution of the Internet of Things direction in 2015-2016 [8], it will be possible to identify the location of all the people and of the objects of everyday use. That being said, the fact that every object around us is endowed with information and a variety of sensors. This enables obtaining the necessary information in real-time mode about its conditions and the surrounding objects state.

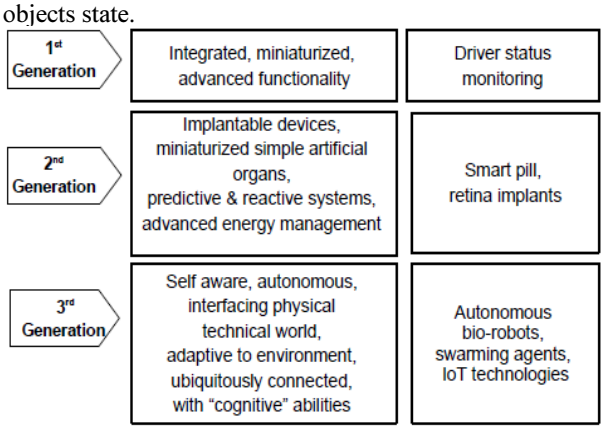


Figure 1. Continuing revolution of Smart System Integration

Today, there are prototypes of Smart Systems that have reached the state of commercial products. In Fig. 1, three generations of Smart Systems integration are presented [9].

III. SMART SYSTEM GEO-CODING PROBLEM

To solve the problem of smart space and its subspaces subjects search, two options can be considered.

For the search world subjects in any space and their subspaces, there are two solutions, namely, (i) geo-coding, and (ii) coordinates determination. Geo2Tag platform supports only subjects geo-markup in a certain area (map); this makes it possible only to search subjects in a given space. In order to be able to markup and find subjects not only in a given space, but also in its subspaces, e.g., in the space of a building, city, street, etc., it should be possible to determine the coordinates of subjects in the predetermined space or their markup by a predefined plan of space subjects, such as subspace ontology, map ontology, etc. This method can also be used to determine the coordinates of the moving objects.

To resolve this problem within a smart space it is required to develop a special knowledge processor (KP); by composing a subspace ontology, it creates a semantic representation model. Coordinates of each subspace subject are obtained from Geo2Tag platform after its labeling on pre-created visualization subspace.

In other words, the subspace is created using a special visualization technology [10]; then, all subjects are placed into this subspace. These subjects are marked within a special representation subspace map (all markup subjects are assigned the coordinates), leading to a representation ontology of the given subspace. KP ontology is used for processing and obtaining information from the given subspace.

Fig. 2 shows the geo-location approach on the example of the Smart-M3 platform. The main Geo-location KP is designed to handle all the information from the geo-space. Ontology Space Creation KP will be used for the building subspaces from its ontology.

Context management system KP server for convenient context management processed by all (sub)spaces. Also, the

 ISBN: 978-1-61208-318-6

approach includes various space sensors, KPs for processing coming information from the sensors, and also the representation and description space ontologies with its domains.

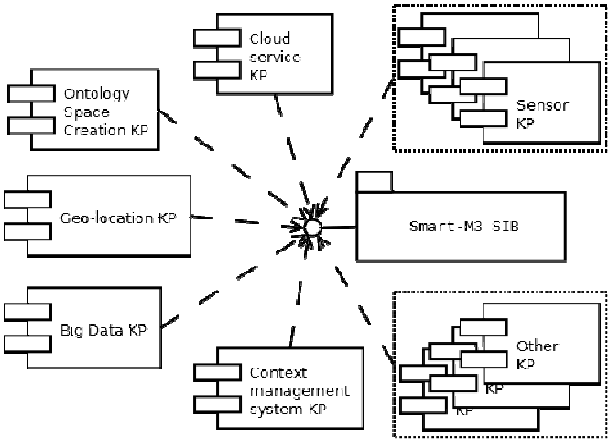


Figure 2. Geo-location Smart-M3 approach

This scheme could be extended by adding knowledge processors for off-line data processing.

Further on, we will present the geo-location agent used to integrate Smart-M3 and Geo2Tag platforms for processing subjects' coordinates in the space.

IV. GCSS SYSTEM REQUIREMENTS

The first main task of the Smart System platform is the integration of Smart-M3 and Geo2Tag platforms, and also expanding the smart space with new data, e.g., geo-data [10]. There are several promising use-cases of Geo-Coded Smart Space (GCSS):

- Geographical markup of smart space data;
- Search set reduction;
- Search context rectification.

GCSS should implement main features from both types of platforms, which are:

- Providing interfaces for semantic data and access;
- Smart-M3 API – Qt [11], Python [12], Java [13];
- Distributed storage for semantic information;
- Interfaces for association semantic objects with geo-tags;
- Spatial and temporal filtration.

Also non-functional requirements should be taken in account:

- Performance – ability to work with big amount of semantic objects geo-tags like cloud based massive offline processing and local context indexing/caching.
- Compatibility – the GCSS should be accessible by legacy interfaces (i.e., SSAP or REST), which is required for seamless integration with existing systems.

Below is the list of the main functional use-cases of the integration platforms agent:

- Smart space Smart-M3 platform management (leave, join, query, insert, delete, update, subscribe, unsubscribe);
- Geo2Tag platform management (connect, disconnect, obtain platform data, search, filtration);
- Geo-tags conversion mechanism to space data (triples) and vice versa;
- Smart space searching and filtering algorithms by means of Geo2Tag platform;
- Ranking mechanism of space data (the algorithm of selection the latest objects by location, optional);

The last three use-cases are fulfilled by the main features of the agent to increase the space with new information, i.e., geo-data; that will be used to determine the location and search for objects in space. The first two are available on Smart-M3 and Geo2Tag platforms.

V. GCSS HIGH-LEVEL DESIGN

High-level layered design for GCSS is presented in Fig. 3. Each level of the system is responsible for the functions and includes its own interface. The following are the layers of the system GCSS:

Interfaces level is responsible for data representation and processing for applications and services;

Integration level contains components for translating geographical data from Geo2Tag format to Smart-Space format and vice versa;

Domain engines level contains particular implementations of smart-space and geo-coding middleware;

Data cloud backend – optional components, which is being used for providing advanced services like off-line data pre-processing, storage for Binary Large Objects, indexing, caching, etc.

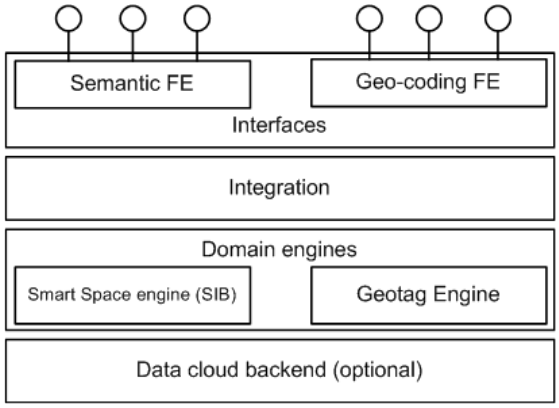


Figure 3. High-level layered design of GCSS

There are five basic components (levels) [14] that provide basic functioning contour of the system (system life cycle):

Data acquisition level – presented by sensors and other receiving information interfaces from the outside world, a person, other systems;

Data pre-processing level – data storage and transformation of the primary form to a form suitable for analysis and decision–making;

Decision-making level – module responsible for information processing and making decisions to achieve the goals of the system, and support tasks related to self-diagnosis and self-organization;

Command level – responsible for making the transformation into control signals own functional components and external systems for the environmental impact implementation;

 ISBN: 978-1-61208-318-6

Action level – implementation of information and physical control of external systems, including the task of encoding and transmitting control signals to run-time systems and control command execution.

The main object of the platforms integration is the integration agent or mediator. Its primary task is to provide interaction between Smart-M3 and Geo2Tag platforms and the platforms data conversion into one common format (triplets). Each platform has the necessary programming interface (API).

VI. PLATFORM INTEGRATION AGENT

The integration agent (GCSS) is responsible for the platforms integration and fills the Smart-M3 space with geo-data by conversion mechanism. Next, the agent will combine the functionality of both platforms (Smart-M3 and Geo2Tag) and will become a sort of common platform within the device to control and manage data between all smart space devices [15].

One can create a agent ontology by using special Smart-M3 ontology generator, i.e., Smart Slog [16][17].

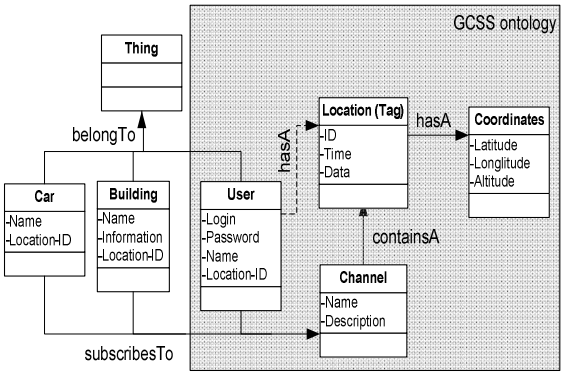


Figure 4. Overlay ontology used by GCSS

GCSS ontology consists of four classes – User class, Channels class, the Tag itself, and its Coordinates. Class User is responsible for a user's of the Geo2Tag platform in space, the tags channel describes a set of tags for a given criterion, the Tag class describes itself data. It should be noted that users can subscribe to an unlimited number of channels, as well as a channel can contain unlimited number of tags. Class User can directly communicate with the tag through the property hasA. Coordinates are allocated in a separate class for more convenient their representations in an agent ontology. More details on GCSS ontology are described in [18].

Each Geo2Tag platform user, if it exists, will be associated with its own user in the Smart-M3 space; if not, then, a new space user will be created; this will be automatically attached to the tag location and to the channels which it subscribes to. Location may be attached to any space object after adding new property (e.g., Location-ID) in the object class of the space ontology. Class Tag property Data is mainly used for searching and filtering space objects, but it can be also used for its association with the object.

It should be noted that the user location or other space object (not static) can change location with time and in order to remain relevant data necessary to provide handling this situation. Smart-M3 platform provides a publisher-subscriber mechanism; by subscribing to specific triplets, the object will automatically receive new data after each change. In our case, these data are the properties of the Coordinates class.

The agent will use the object model of the ontology representation, i.e., have clearly documented ontology classes names and their properties, as well as certain triplets (subscription). Thus, the space agent ontology will look like a list of properties that are linked by a predicate. In the first version of the integration agent, the space will be filled only with geo-data, which will be linked with their space objects (a person, object, etc.). In the future, we plan to expand the space by the addition of the users and channels tags information.

All Geo2Tag platform data are stored in a database on a dedicated server. Geo2Tag platform allows recording and retrieving data using Representational State Transfer (REST) specific queries [19] in Java Script Object Notation (JSON) format [20]. There is also a variety of clients to work with a Geo2Tag platform, mainly for mobile platforms.

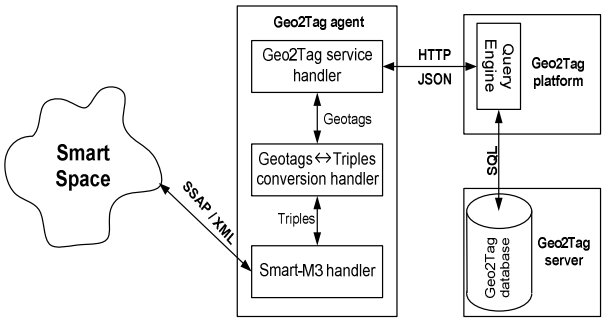


Figure 5. GCSS architecture

The integration with the Smart-M3 platform will be implemented through a special mediator (agent). Its main task is to convert data from one platform format (Geo2Tag, JSON) to another format (Smart-M3, XML). As mentioned above, the Geo2Tag platform transmits data in JSON format; this is a text format, but in a more readable form for humans.

The agent consists of three main components:

- Geo2Tag service handler;
- Geotags – Triples conversion handler;
- Smart-M3 handler.

Geo2Tag service handler is responsible for obtaining geo-data, it connects to the server database and requests data using a special class LoadTagsQuery.

Geotags–Triples conversion handler is required to bring data to a convenient form for the triplets creation. Since the data are returned in JSON format, they need to be parsed and pulling the necessary data, namely, time, location and description of the geo-tag by saving them for later processing.

At the last stage, it is a connection to the space; then, triplets are created, according to the ontology. Finally they placed into the Smart-M3 space. Below, a list of the location triplets that are created during conversion mechanism is presented:

- < User, “hasA”, Location-ID >
- < Location, “hasID”, ID >
- < Location, “hasTime”, Time >
- < Location, “hasData”, Data >

 ISBN: 978-1-61208-318-6

- < Location, “hasLatitude”, Latitude >
- < Location, “hasLongitude”, Longitude >
- < Location, “hasAltitude”, Altitude >

The main evaluation criteria of the Smart System platform will serve for its performance, the ability to integrate into embedded devices, the amount of transmitted traffic, and the response speed. The analysis should show how the platform behaves in the real conditions and only then takes steps to improve its operability.

VII. GEO2TAG AND SMART-M3 INTEGRATION

One of the main action of the integration agent use-case is the geo-tags conversion mechanism to the space triplets; below the pseudo-algorithm is presented;

Connect to the Geo2Tag platform by using a *Login()* query;

Point service (database) by *setDB()* query, where data will be obtained;

Sampling nearest tags with a *LoadTags()* query or *Filter()* inside the defined geometry figure;

Obtaining the necessary tags parameters from received data (JSON format);

Formation of the initial triplets for space objects representation by class Triple(S, P, O): a triplet for linking space object with its location, a triplet for location time, coordinates and data. In general, six triplets describe *Location (Tag)* of space object.

Connection and insertion triplets in space are done with the help of Smart-M3 API.

After the execution of the algorithm, the space will be filled with latest tags from the Geo2Tag server database. The inverse transform mechanism (triplets to the geo-tags) are quite similar, but only performed at the Smart-M3 platform.

Now, the integration agent responsible for the platforms integration fills the space (Smart-M3) with geo-data by conversion mechanism. Next, the agent will combine the functionality of both platforms (Smart-M3 and Geo2Tag) and will become a type of common platform within the device to control and manage data between all smart space devices.

The next main platforms integration agent use cases are smart space data searching and filtering algorithms.

A filtering mechanism of space data is required to obtain relevant information at the moment when the system works; therefore, by filtering the objects by location, we will have a list of the most relevant data at a given time. We consider a filtering data mechanism based on their metadata obtained by SparQL queries [21].

Each ontology object has a set of metadata, for example, *Id*, *Description*, *Type*, *Time*, *Position*, *Status* (e.g., *Offline, Online, Connecting*). Object metadata is used in the filtering process to retrieve only those objects that satisfy the consumer (client) requirements.

Searching and filtering algorithms are based on the Geo2Tag platform is filtering queries; as a result, as SparQL queries require significantly more Smart-M3 resources, it might affect to the performance of the whole system.

A general Smart Space data filtering algorithm scheme based on the Geo2Tag platform via Smart-M3 is presented in Fig. 6.

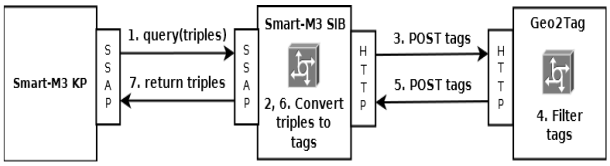


Figure 6. General Smart Space data filtering algorithm scheme

This filtering method operates as follows: KP sends a request to the Smart-M3 for sample required triplets; then, Smart-M3 makes a request to the Geo2Tag for retrieval necessary data, for example, by Radius [22]; Smart-M3 converts the triplets to tags and sends them back to KP.

After the development of the Smart System platform within the device, each of its mechanisms (algorithm) will be subject to thorough analysis by the following criteria: universality, performance, resources, the ability to integrate into embedded devices, memory size, the amount of transmitted traffic, response speed [23]. We expect that the analysis will show how the platform behaves in the real conditions, to improve its operability.

VIII. CONCLUSION

In this paper, we proposed a description of a Smart System based on common device platforms; we discussed the requirements and use-case of platform systems, its high level design and architecture for smart-space and geo-coding middleware integration. This integration could be made by using special Smart-M3 Knowledge Processor, which monitors both spaces and translates data from one to another and vice versa.

The current results of the project:

- Integration platforms agent prototype;
- Geo-tags conversion mechanism;
- Filtering mechanism based on the platform Smart-M3.

The next step in the development of Smart Systems device platform is the complete platform components integration, common protocols and interfaces for communicating between all devices. There are still open questions for future development: overall system performance, effective object monitoring, temporal and spatial filtration, integration with media objects.

ACKNOWLEDGMENT

The authors would like to thank Finnish Russian University Cooperation in Telecommunication Program for provided equipment and JetBrains Company for financial support.

REFERENCES

[1] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey”, Communications Surveys Tutorials, IEEE, 2013 pp. 1–44.

[2] N. Nabian, C. Ratti, A. Biderman, and G. Grise, “MIT GEOblog: A platform for digital annotation of space for collective community based digital story telling,” 3rd IEEE International Conference on Digital Ecosystems and Technologies, Piscataway, N.J., IEEE, 2009, pp. 353-358.

[3] J. Rishede, T. Man, and L. Yiu, “Effective Caching of Shortest Paths for Location-Based Services”, SIGMOD ’12, Scottsdale, Arizona, USA, 2012, pp. 313-324.

[4] K. Kolomvatsos, V. Papataxiarhis, and V. Tsetsos, “Semantic Location Based Services for Smart Spaces,” 2nd International

 ISBN: 978-1-61208-318-6

Conference on Metadata and Semantics Research (MTSR), 2007, Corfu, Greece, pp. 515-525.

[5] D. J. Cook and S. K. Das, "How smart are our enviroments?" an updated book look at the state of the art, Pervasive and Mobile Compution 3(2), 2007, pp. 53-73.

[6] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, "Smart-M3 Information Sharing Platform", 1st Workshop on Semantic Interoperability in Smart Spaces, 2010, pp. 1041-1046.

[7] I. Bezyazychnyy, K. Krinkin, M. Zaslavskiy, S. Balandin, and Y. Koucheravy, "Geo2Tag Implementation for MAEMO", 7th Conference of Open Innovations Framework Program FRUCT, 2010, Saint-Petersburg, Russia, pp. 7-11.

[8] http://www.compression.org/energy-productivity-of-systems/, [retrieved: Jan, 2014].

[9] EPoSS Strategic Research Agenda 2009 – http://www.smart-systems-integration.org/public/documents/publications/, [retrieved: Jan, 2014].

[10] M. Nollenburg, "Geographical visualization", Human-Centered Visualization Environments, Lecture Notes in Computer Science, vol. 4417, 2007, pp. 257-294.

[11] http://qt-project.org/, [retrieved: Jan, 2014].

[12] http://www.python.org/, [retrieved: Jan, 2014].

[13] http://www.java.com, [retrieved: Jan, 2014].

[14] G. Akhras, "Smart Materials and Smart Systems for the future", Canadian Military. Journal 2000, pp. 25-31.

[15] D. Korzun, I. Galov, A. Kashevnik, N. Shilov, K. Krinkin, and Y. Korolev, "Integration of Smart-M3 Applications: Blogging in Smart Conference," Proc. 4th Conf. Smart Spaces (ruSMART 2011), Saint-Petersburg, Russia, 22-23 August 2011, pp.51-62.

[16] D. Korzun, A. Lomov, P. Vanag, J. Honkola, and S. Balandin, "Generating Modest High-Level Ontology Libraries for Smart-M3", Proc. 4th Int'l Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM, 2010, pp. 103–109.

[17] D. Korzun, A. Lomov, P. Vanag, J. Honkola, and S. Balandin, "Multilingual ontology library generator for Smart-M3 information sharing platform", International journal on Advances of Intelligent System 4 (3&4), 2011, pp. 68-81.

[18] K. Krinkin and K. Yudenok, "Geo-coding in Smart Environments: Integration Principles of Smart-M3 and Geo2Tag," In Proceedings of the 13th International Conference, NEW2AN 2013 and 6th Conference, ruSMART 2013, St. Petersburg, Russia, August 28-30, 2013, Proceedings. Springer 2013 Lecture Notes in Computer Science, pp. 107-116, ISBN 978-3-642-40315-6.

[19] http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm, [retrieved: Jan, 2014].

[20] http://www.json.org/, [retrieved: Jan, 2014].

[21] E. Nageba, P. Rubel, and J. Fayn, "Semantic agent system for automatic mobilization of distributed and heterogeneous resources," In Proceedings of the 3rd International Conference on Web Intelligence, Mining and Semantics, WIMS '13,ACM, New York, NY, USA, 2013, Article 28, pp. 9-17.

[22] http://geo2tag.org/index.php/Exchange_protocol, [retrieved: Jan, 2014].

[23] M. Zaslavsky and K. Krinkin, "Geo2tag Performance Evaluation," Proceedings of the 12th Conference of Open Innovations Association FRUCT and Seminar on e-Travel, Oulu, Finland, 2012, pp. 185-193.

 ISBN: 978-1-61208-318-6

OpenFlow Networks with Limited L2 Functionality

Hiroaki Yamanaka, Eiji Kawai, Shuji Ishii, and Shinji Shimojo
Network Testbed Research & Development Promotion Center
National Institute of Information and Communications Technology
KDDI Otemachi Bldg. 21F, 1-8-1 Otemachi, Chiyoda-ku, Tokyo 100-0004, Japan
Email: {hyamanaka, eiji-ka, shuji, sshinji}@nict.go.jp

***Abstract*—OpenFlow enables flexible control of network traffic with arbitrary flow definitions. On carrier access networks, OpenFlow can be used to provide customized network settings for each client for virtual private network (VPN), Internet Protocol television (IPTV), and content delivery network (CDN) services, etc. Since there is a massive amount of traffic in carrier access networks, high performance switches are necessary. However, costs tend to increase due to the number of switches. An OpenFlow switch processes wide-range of header fields and supports wildcard matching. Large spaces for ternary content addressable memory (TCAM) and application-specific integrated circuits (ASICs) are used for high performance lookups. Currently proposed techniques reduce the amount of energy that is consumed by reducing the frequency of TCAM usage in switches. However, these techniques require complex functionality in order to manage matching in the switches. As a result, switches are still expensive. In this paper, we propose a technique that enables the construction of OpenFlow networks using switches that require little more than L2 switch functionality. The functionalities that are required for the switches include an OpenFlow interface for handling the flow table externally and a simple matching function for the MAC header. Arbitrary flow definitions from an OpenFlow controller are translated to the flow definitions by the MAC address at the external proxy.**

Keywords-OpenFlow; TCAM; L2 switch; carrier access network

I. INTRODUCTION

Software Defined Networking (SDN) technologies (e.g., OpenFlow [1]) ease network management, service management, and quality of service (QoS) provisioning. SDN technologies are being considered for use in carrier-grade networks. One possible model for applying SDN technologies in carrier networks involves the control of fined-grained flows using OpenFlow in carrier access networks and multi-protocol label switching (MPLS) tunnels (i.e., relatively simple logic for packet forwarding) in core networks [2].

When OpenFlow is deployed in carrier access networks, the costs for infrastructure are relatively high. There are two reasons for high costs. The first is that there are many OpenFlow switches in carrier access networks. The second reason is that hardware OpenFlow switches are costly. The ternary content addressable memory (TCAM) significantly increases the costs that are associated with a hardware-based OpenFlow switches. TCAM is a special type of memory that enables matching on the headers of received data packets during one clock cycle, regardless of the number of entries in memory. This capability of TCAM is preferable in carrier access networks because it enables high performance for the forwarding of high volumes of data packets. However, TCAM is power hungry and expensive [3]. It has been noted that TCAM is up to 80 times more expensive than static random access memory (SRAM) [4]. In an OpenFlow switch, the required TCAM space is large due to the wide range of header fields that are supported in OpenFlow [5].

Techniques have been proposed in the research community for reducing the amount of power that is consumed by TCAM in an OpenFlow switch. The basic idea is to decrease the frequency of the usage of TCAM. Only the first data packet in a flow is matched using TCAM and subsequent data packets that have the same header fields are matched using SRAM or binary content addressable memory (BCAM). In DevoFlow [6], SRAM is used in conjunction with the hash method in order to match the subsequent data packets. The hash method improves the performance as far as possible when SRAM is used. Generally, when the TCAM is avoided in a switch, it is necessary to include chipsets in the switch for complex packet processing (e.g., applying the hash function) in order to obtain high performance packet forwarding.

In this paper, we propose a technique that retains the high performance of during packet forwarding and lowers the cost of switches for OpenFlow infrastructure through the use of relatively simple and inexpensive devices. The idea is to restrict the matching fields in the memory of a switch to the source MAC header and allow the switch to perform matches in a simple manner using the single source MAC header. Meanwhile, the proposed technique enables an OpenFlow controller and the end-hosts to use all of the header fields that are supported in OpenFlow. The external proxy that is between an OpenFlow controller and the switches translates the matching fields that were originally defined by the controller to the matching fields for the source MAC addresses. Furthermore, the proxy manages the edge switches in order to modify the source MAC addresses for data packets and forward them to the network. In the network, the source MAC address of a data packet represents all of the original header fields. As a result, the switches in the network only need to match on the single source MAC header. The proposed technique enables the construction of an OpenFlow network with switches that are implemented using similar chipsets of L2 switches.

The remainder of this paper is organized as follows. Section II describes the mechanism and the limitations of

 ISBN: 978-1-61208-318-6

the current technique for reducing TCAM usage. Section III describes the concept and the architecture of our proposal. Section IV describes the detailed implementation of the proposal. Section V evaluates the overheads that are associated with the proposal. Section VI contains remarks about related work and Section VII presents the conclusion.

II. Reducing TCAM Cost in OpenFlow

A typical hardware-based OpenFlow switch contains TCAM for high performance data packet processing in a network. An OpenFlow [1] network is composed of a controller and a group of OpenFlow switches. The controller and the OpenFlow switches communicate through a control plane on the network in order to maintain flow entries in the switches. The OpenFlow switches transfer data packets on the data plane of the network based on their flow entries. A flow entry includes definitions of the flows that are referred to as the matching fields. The matching fields include the ingress port number and the header fields from layers 2–4 that are specified in the OpenFlow switch specification [5]. Wildcards are allowed for any of the matching fields. An OpenFlow switch searches the flow entries that need to be matched in the header fields of each data packet that is received. TCAM enables searching during one clock cycle, regardless of the number of the entries in the TCAM and regardless of whether wildcards are included or not included in the matching fields.

Since TCAM is power hungry and expensive, it increases the infrastructure costs for OpenFlow networks tremendously. State-of-the-art techniques have been proposed academic papers in order to reduce the frequency of the usage of TCAM (i.e., energy consumption) in switches. These techniques allow TCAM to only be used for matching for the first data packets that arrive, while subsequent data packets are matched using SRAM or BCAM. A switch sees all of the header fields for the first data packet that are matched using TCAM. Then, it sets up the matching fields for the subsequent data packets using SRAM or BCAM. When SRAM is used, matching can be implemented using the hash method for subsequent data packets in a constant time, which is not one clock cycle of central processing unit (CPU). When BCAM is used, matching can be implemented for subsequent data packets in one clock cycle of CPU.

In the section below, *"wildcard matching fields"* refers to matching fields in which at least one header field is a wildcard. For an IP header, it may be the IP prefix. *"Exact matching fields"*, on the other hand, refers to matching fields in which there is no header field with wildcards or IP prefixes.

A. Setting Exact Matching Fields

This section summarized the method for reducing the frequency of TCAM usage. This method is found in DevoFlow [6]. This method determines the exact matching fields inheriting from the wildcards using the header fields of the data packets that are arriving at the switch. The procedure for setting the exact matching fields and the data packet processing is as follows (Figure 1).

1) The wildcard matching fields that are originally set by the OpenFlow controller are memorized in TCAM in the switch.

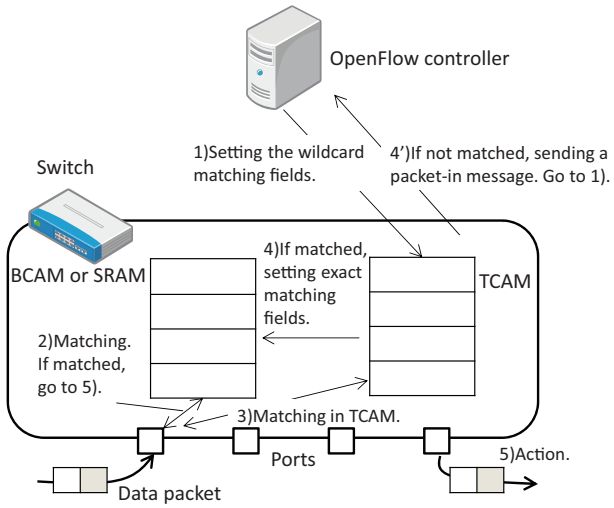


Figure 1. The basic procedure of setting exact matching fields.

2) When a data packet arrives, its header fields are matched to the exact matching fields in BCAM or SRAM. If the header fields are matched, the data packet is processed based on the flow entry (go to 5)).
3) If they are not matched to any of the exact matching fields in memory, then the header fields are matched to the wildcard matching fields in TCAM.
4) If the header fields are matched using TCAM, then the corresponding exact matching fields (Figure 2) are stored in BCAM.
4′) If they are not matched, then the data packet is forwarded as a packet-in message to the OpenFlow controller in order to query about the proper method for processing the data packet.
5) The data packet is processed using the action that is specified in the matching flow entry.

B. Limitations

Generally, there is a trade-off between the level of packet forwarding performance and the complexity of the chipsets for switches when TCAM is not used. Even if it is possible for a switch to process only the exact matching fields, BCAM is still necessary in order to obtain line-rate performance for packet forwarding. Because there are fewer circuits in BCAM than in TCAM [3], the prices for BCAM devices are lower and the devices also consume less energy. However, BCAM is still more costly than SRAM. Current techniques propose methods for obtaining high performance levels with limited BCAM space.

In DevoFlow [6], the exact matching fields are stored in SRAM and the hash method is used to search the flow entries that need to be matched for a data packet. The chipset for DevoFlow is relatively simple. However, the performance is limited because the matching process largely depends on the CPU.

Congdon et al. [7] utilized BCAM to match against the exact matching fields. BCAM stores the small size data of a partial header field or a hash value of the exact matching

 ISBN: 978-1-61208-318-6

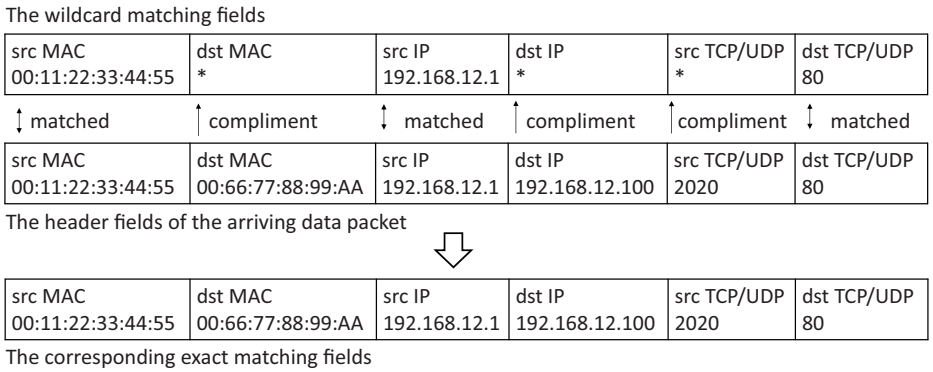


Figure 2. An example of the wildcard matching fields and the exact matching fields.

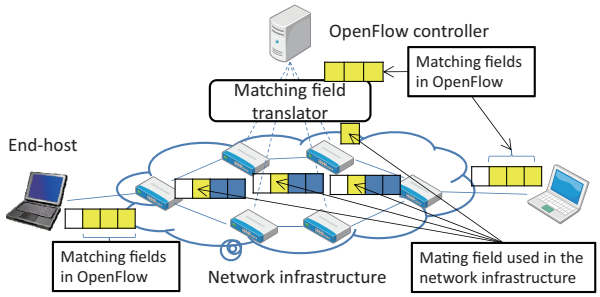


Figure 3. Limiting the matching fields in the network infrastructure.

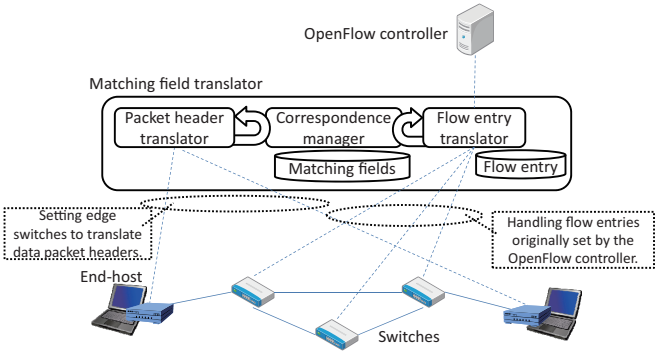


Figure 4. The architecture of the matching field translator.

fields. Along with the small data, BCAM stores the pointer to the original matching fields stored in SRAM. When the partial value or the hash value of the data packet header is matched to the data in BCAM, the correctness of the matching is confirmed by referencing the original exact matching fields in SRAM. This method requires a chipset that is capable of performing a certain amount of complex logic.

III. Proposal of OpenFlow Deployment using the L2-based Switches

We propose a technique that enables deployments of OpenFlow networks using simple, low-cost switches. The main idea of this technique is to limit the use of matching fields to the single source MAC header inside the network (Figure 3). Switches inside the network simply match on the source MAC header fields of the data packets. Meanwhile, the technique enables an OpenFlow controller and the end-hosts to use all of the matching fields that are normally supported in OpenFlow as the matching fields. As a result, this technique retains the programmability of OpenFlow for an OpenFlow controller. Because the single MAC header is the only matching field for single flow entries in a switch, more flow entries can be stored in BCAM. Furthermore, the switch simply matches on the MAC header. As a result, the chipset for the switch requires only minor extension beyond what is required in an L2 switch.

A. Summary of the proposed technique

The technique maps the matching fields that the OpenFlow controller and end-host manage to the corresponding MAC addresses that are manged by switches inside the network. The matching fields for the OpenFlow protocol messages are mapped for the OpenFlow controller. The header fields of data packets are mapped for end-hosts. In the section below, we refer to the translated MAC addresses that are managed by switches inside the network handle as *MAC ID address*.

There is a manager for the correspondence between the exact matching fields and the MAC ID addresses. The correspondence is global throughout the network. Based on this correspondence, the exact matching fields that are specified by the OpenFlow controller are translated into the source MAC ID addresses and the source MAC ID addresses are stored in the BCAM of the switch. For the wildcard matching fields, the exact matching fields are determined using the header fields in actual data packets that arrive, and then, the exact matching fields are translated into the MAC ID addresses.

At a network ingress switch, the source MAC header in the data packet is replaced with the MAC ID address, which corresponds to all of original header fields in the data packet. At a network egress switch, the original header fields from the data packet are recovered based on the correspondence with the source MAC ID address in the data packets and the data packet is transferred to the end-host.

B. Architecture

There is a proxy between the switches and the OpenFlow controller that is referred to as a “matching field translator”. For the OpenFlow controller, the matching field translator behaves like an OpenFlow switch. For the switches, the

 ISBN: 978-1-61208-318-6

TABLE I. AN SIMPLE EXAMPLE OF THE CORRESPONDENCE OF THE ORIGINAL MATCHING FIELDS AND THE MAC ID ADDRESSES.

Original L2–L4 headers			MAC ID address
L2	*L3*	*L4*	
00:11:BB:CC:DD:EE	192.168.1.1	80	00:00:00:00:00:01
00:11:BB:CC:DD:EE	192.168.1.1	22	00:00:00:00:00:02
00:01:BB:CC:DD:FF	192.168.1.100	80	00:00:00:00:00:03

matching field translator behaves like an OpenFlow controller. As shown in Figure 4, there are the correspondence manager, the flow entry translator, and the packet header translator in the matching field translator.

1) The correspondence manager: The correspondence manager handles the correspondences between the exact matching fields and the MAC ID address. For the exact matching fields, the correspondence manager allocates a 48-bit ID in the form of a MAC address. Table I shows a simple example of the correspondence between the matching fields and the MAC ID addresses. The matching fields and the IDs are in a one-to-one correspondence. There are two modules that search the correspondence: the flow entry translator and the packet header translator. If the correspondence manager has not allocated IDs to the matching fields yet, it allocates a new ID and returns it as the MAC ID address. This ID allocation can be implemented by allocating sequential numbers to the new exact matching fields. Searching can be implemented using the hash method in a constant time.

Note that the MAC ID address always serves as the basis for the matching fields that are specified by the OpenFlow controller or for the header fields in the data packets that are sent and received by end-hosts. As a result, when the original matching fields are searched using the MAC ID address, the corresponding matching fields always exist.

2) The flow entry translator: The flow entry translator modifies and relays the OpenFlow protocol messages between the OpenFlow controller and the switches. For a flow entry installation for the exact matching fields, the flow entry translator simply replaces them with the corresponding MAC address and forwards the message to the switch.

For a flow entry installation for the wildcard matching fields, the flow entry translator determines the corresponding exact matching fields when the new data packet arrives at the switch. Then, the flow entry translator obtains the corresponding MAC ID address, and sends the flow entry installation message for the MAC ID address.

For a flow statistics request, from the OpenFlow controller, for the flow entry of the wildcard matching fields, the flow entry translator collects the flow statistics from the switch, at first. The objectives of the collection are the flow entries of the exact matching fields that correspond to the wildcard matching fields. Then, the flow entry translator responds with the aggregated value as the statistics for the requested flow entry.

3) The packet header translator: The packet header translator modifies the header fields of data packets that are transferred through the edge ports (i.e., the ports that connect directly to the end-hosts) of the edge switches. For a data packet that is sent from an end-host, the packet header translator simply replaces the source MAC header with the MAC ID address corresponding to the original header fields in the data packet. For a data packet that is sent from the switch to the end-host, all of the header fields are set to the values in the exact matching fields that correspond to the source MAC ID address of the data packet. Note that header fields other than the source MAC header are modified at an egress switch because the OpenFlow controller may have modified other header fields.

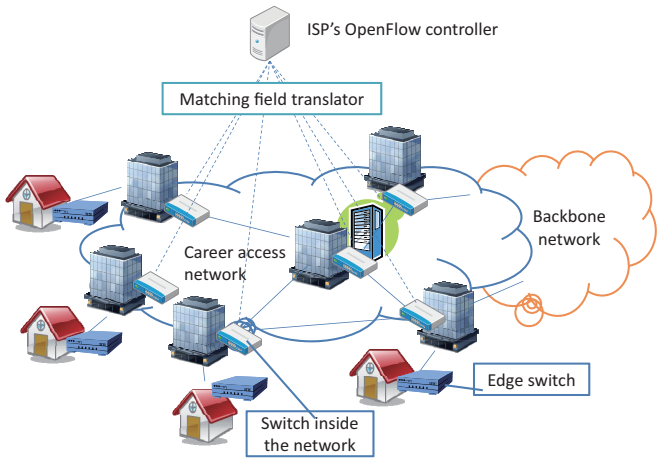


Figure 5. Deployment in a carrier access network.

C. Deployment Scenario

The proposed architecture can reduce the capital expenditure (CAPEX) and operational expenditure (OPEX) for infrastructure in a carrier access network deployment scenario [2]. On the carrier access network, an ISP's controller manages flows for clients in order to provide customized networks settings for virtual private network (VPN), Internet Protocol television (IPTV), content delivery network (CDN) services, etc. As shown in Figure 5, an ISP's controller manages switches in central offices and data centers in the carrier access network via the matching field translator. Edge switches for packet header translation exist in each homes. Although the edge switches need to match all of the header fields, the cost of edge switches is not high. Since the amount of traffic is relatively low in individual home, software-based switches are sufficient for meeting the demands that are replaced on edge switches.

IV. DETAILED IMPLEMENTATION

In this section, we describe the detailed implementation of packet header translation at ingress and egress switches, and the handling of flows (i.e., translations for flow entry installations, obtaining flow statistics, and managing packet-out messages). These scenarios involve translations of the matching fields into the MAC ID addresses.

A. Translation of data packet headers

Edge switches rewrite the headers of data packets that are sent directly from and to end-hosts. The edge switches include OpenFlow functionality, i.e., they can manage all of the supported header fields. An edge switch has at two ports. One is connected to an end-host in a home network. The other is

 ISBN: 978-1-61208-318-6

connected to the central office, i.e., the carrier access network. The packet header translator is configured in advance with information that specifies the port that connects to the end-host or the central office.

When an unknown data packet (i.e., no matching flow entry is set) arrives at the port in the edge switch that connects to an end-host, the switch sends a packet-in message to the packet header translator. Then, the packet header translator obtains the corresponding MAC ID address for the header fields in the data packet. Finally, the packet header translator sets the flow entry in the edge switch. The matching fields for the flow entry include the header fields of the data packet. The actions that are performed include modifying the source MAC header to the corresponding MAC ID address and sending out the data packet from the other port. Subsequent data packets for the same header fields are simply matched. Then, flow entry actions are applied and the data packets are transferred.

When an unknown data packet arrives at the port of an edge switch that connects to the central office, the switch obtains the flow entry from the packet header translator. The flow entry is used for recovering the original data packet header and transferring the data packet from the other port. The matching fields in the flow entry include the source MAC ID address, which corresponds to the original header fields. The actions that are performed include the modification of all of the header fields to the original header fields. Note that the OpenFlow controller may modify arbitrary header fields in data packets in the network. Since only the source MAC header is managed in the network, other header fields are left unchanged. However, the source MAC ID address of the arriving data packet corresponds to the header fields when the data packet arrives at the switch. This is guaranteed by the translation of the flow entry installation that is described in Section IV-B2.

B. Flow Entry Installation

In order to manage flows, the flow entry translator manages the correspondence of the original flow entries and the exact matching fields for the flow entries that have actually been installed (Table II). The original flow entries are for the wildcard or exact matching fields that were set by the OpenFlow controller. The installed flow entries are for the exact matching fields, which are the same as the original exact matching fields or the exact matching fields that were specified by the data packet that actually arrived. The flow entry installation process is described in the section below. In this process, the correspondence database (i.e., Table II) is updated when a flow entry is installed into, or removed from the switch.

For the installation of a flow entry whose matching fields are the exact ones, the flow entry translator simply translates the flow entry whose matching fields are the source MAC ID address, and installs ("FI-iii" in Figure 6). For the installation of a flow entry whose matching fields are the wildcard values, the flow entry translator installs possible flow entries in the switch first in order to preserve the consistency of the priorities in the switch's flow table ("FI-i" in Figure 6). Then, the flow entry translator determines the corresponding exact matching fields ("FI-ii" in Figure 6) and installs the flow entry ("FI-iii" in Figure 6) .

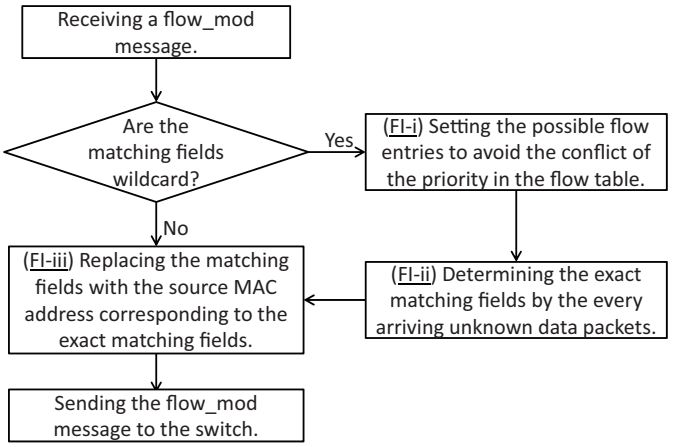


Figure 6. Summarized process of flow entry installation.

1) Maintaining the priorities in the flow table (FI-i): As described in Section IV-B2, for a flow entry for the wildcard matching fields, the flow entry translator does not immediately install the flow entry when it receives the flow entry installation message, i.e., a flow_mod message. Due to the time lag that is associated with the installation of a flow entry, an arriving data packet can be matched inappropriately to another flow entry whose priority is lower than that of the original one.

For instance, in Table II, let us assume that a flow entry for switch 0x1 (the priority is 65536 and the matching fields are "00:11:22:33:44:55, 192.168.10.2, 80") is not installed. Meanwhile, let us assume that another flow entry for switch 0x1 (the priority is 65534 and the matching fields are "00:11:22:33:44:55, 192.168.10.2, 80") is installed. Subsequently, a data packet that arrives with header field values of "00:11:22:33:44:55, 192.168.10.2, 80" should be matched to the flow entry with priority 65536. However, the data packet matches to the one with priority 65534, and the switch fails to send a packet-in message. This conflicts with the OpenFlow controller because the OpenFlow controller assumes that the data packet matches to the flow entry with priority 65536.

In order to avoid this conflict, the flow entry translator installs the possible flow entries for the exact matching fields corresponding to the flow entry (denoted by "E") for the wildcard matching fields. First, the flow entry manager compares the wildcard matching fields for E and the exact matching fields for the flow entries that have already been installed for the lower priorities in the flow table. Then, for the installed flow entries that are found, whose exact matching fields overlap with the matching fields of E, the flow entry translator immediately installs the flow entries that correspond to E. The exact matching fields of the installed flow entries are the same as the matching fields for the flow entries that were found. The priority of the flow entries are the same as the priorities for E. The actions for the installed flow entries are the same as the actions for E. The specified flow entries are translated and installed by the same process that is found in "install-iii" in Figure 6.

2) Determining the matching fields and installing the flow entry (FI-ii): The flow entry translator determines the corresponding exact matching fields based on the data packets that

 ISBN: 978-1-61208-318-6

TABLE II. AN EXAMPLE OF THE FLOW ENTRY CORRESPONDENCE DATABASE.

Switch ID	Original matching fields	Priority	Installed exact matching fields
0x1	00:11:22:33:44:55, ∗, 80	65536	00:11:22:33:44:55, 192.168.10.1, 80
		65536	00:11:22:33:44:55, 192.168.10.2, 80
0x1	00:11:22:33:44:55, 192.168.10.2, 23	65535	00:11:22:33:44:55, 192.168.10.2, 23
0x1	00:11:22:33:44:55, 192.168.10.2, 80	65534	00:11:22:33:44:55, 192.168.10.2, 80
0x2	00:11:22:33:44:55, 192.168.10.1, ∗	65536	00:11:22:33:44:55, 192.168.10.1, 23
		65536	00:11:22:33:44:55, 192.168.10.1, 80
0x2	00:11:22:33:44:55, 192.168.10.2, 23	65535	00:11:22:33:44:55, 192.168.10.2, 23

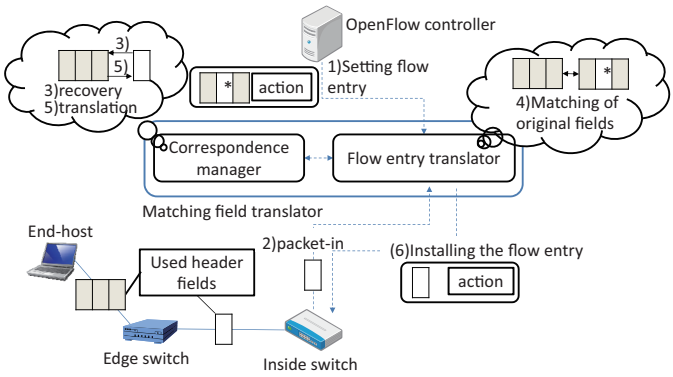


Figure 7. Installing a physical flow entry based on an arriving data packet.

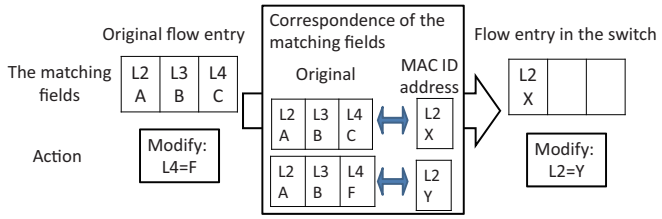


Figure 8. Translation of header modification.

are arriving. When the flow entry translator receives a packet-in message from the switch (#2) in Figure 7), it first recovers the original L2–L4 headers from the source MAC ID address of the data packet (#3) in Figure 7). Then, the flow entry translator compares the L2–L4 headers with the wildcard matching fields from the entries in the flow entry correspondence database (#4) in Figure 7). If there is the flow entry for the wildcard matching fields to be matched to, a flow entry translator generates a flow entry using the exact matching fields that correspond to the source MAC ID address in the packet-in message. Then, the flow entry is translated and installed into the switch by the same process that is described in "install-iii" (#5) and #6) in Figure 7). If there is no flow entry to be matched to, the header fields in the data packet are recovered using the original ones from the source MAC ID address and the packet-in message is transferred to the OpenFlow controller.

3) Translation of a flow entry (FI-iii): For an original flow entry whose matching fields are exact, the flow entry translator simply replaces the original matching fields with the source MAC ID address corresponding to the original matching fields, and installs the flow entry.

When the flow entry translator installs a flow entry into the switch, the action for modifying the header field is also replaced. When an OpenFlow controller modifies the header field of a data packet, only the partial header field that is to be modified is specified in the message. The flow entry translator translates the action field so that only the source MAC ID address is replaced (Figure 8). This translation enables switches to modify only the source MAC header while the source MAC ID address of the data packet in the network always represents the original header fields that should be at the given position in the network at that time. The translated source MAC ID address is the one that corresponds to the combination of the values of all header fields resulting from the modification. The combination of the resulting header fields can be determined by replacing the value of the partial header in the matching fields with the value that is specified in the action of the flow entry.

C. Other processes of flow handling

1) Obtaining flow statistics: If the OpenFlow controller sends a message to obtain the flow statistics from the flow entry for the exact matching fields, the flow entry translator simply replaces the original matching fields to the source MAC ID address. The MAC ID address corresponds to the original exact matching fields.

If the request is for the flow statistics for the wildcard matching fields, the flow entry translator collects the flow statistics from the switch, sums them up, and responds with the sum for the flow statistics from the original flow entry. Since the flow statistics represent the number of data packets that were matched to the flow entry, the sum of the flow statistics for the corresponding flow entries that were installed for the exact matching fields is returned.

2) Packet-out: A packet-out message is for sending out a data packet from the OpenFlow switch. In the message, the data packet that is being sent is specified by the buffer ID or by the data packet included in the message. If the data packet is included in the message, the flow entry translator replaces the source MAC header with the MAC ID address that corresponds to the original header fields in the data packet. Then, the message is forwarded to the switch and the switch sends out a data packet with MAC address that corresponds to the original header fields.

V. EVALUATION

The proposed technique enables simple, low-cost implementations of hardware-based switches for OpenFlow infrastructures. However, the matching field translator adds overheads. The overhead is especially critical for the data transmission performance. The overheads that are caused by the matching field translator are caused by the latencies for flow entry settings. In our proposal, the corresponding matching fields are determined for each new data packet.

 ISBN: 978-1-61208-318-6

In order to evaluate the latency from the flow entry installations, we measured the lost data size due to installations by the flow entry translator. For comparison purposes, we measured the lost data size on a network that was controlled directly by an OpenFlow controller. The parameter that affects the delay of the flow entry installation process is the number of original flow entries for the wildcard matching fields that are set by the OpenFlow controller. The number of original flow entries for the wildcard matching fields directly affects the delay because the matching to the wildcard matching fields is implemented by software, i.e., linear searching.

A. Settings

We assumed a simple network whose data plane was composed of two end-hosts, two edge switches, and a switch. On the control plane, there was an OpenFlow controller and a matching field translator. There was no matching field translator and the software edge switches functioned as the switching hubs when measurement were being performed in the network without the proposed technique. The OpenFlow controller and the matching field translator ran on systems with an Intel Xeon E5520 processor and 6GB RAM. The edge switches used systems with Intel Celeron Dual-Core T3330 processors with 2GB RAMs that ran Open vSwitch [8]. We used the USB Network Interface Cards (NICs) as the ports of the edge switches. The end-hosts ran on systems with Intel Celeron Dual-Core T3330 processors and 2GB RAMs. In the experiments, the operating system on the all hosts was Ubuntu 12.04 LTS.

The switch inside the network was NEC PF 5240 switch. Based on the switch specification that we confirmed, the switch supported a maximum buffer size of 544 packets, i.e., the maximum number of new data packets with the same header fields that can be buffered in the switch is 544. Note that our proposed technique does not require TCAM in a switch, but does require the BCAM space for the source MAC headers. Unfortunately, we did not have a hardware-based switch that incorporates the required and necessary functionality for the proposed technique. As a result, we used the hardware-based OpenFlow switch that includes TCAM when we ran the matching field translator. The exact matching fields of the source MAC ID addresses for the flow entries installed by the matching field translator were processed using TCAM. However, since the matching performance of TCAM and BCAM is same for the exact matching fields, the switch used in this experiment did not affect the results in terms of the packet forwarding performance of a switch in the network.

The OpenFlow controller set the number flow entries for the wildcard matching fields when it connected to the switch. When we used the matching field translator, the flow entries were stored in the matching field translator at first. However, if we did not use the matching field translator, the flow entries were installed immediately into the switch.

We used iperf to send data packets between the end-hosts. The UDP packets were sent at a rate of 75.40 Mbps because that is the maximum throughput for environment that was used in the experiment. The maximum throughput was low due to the USB NICs. However, it was sufficient because we assumed that the edge switches were home router-class switches. In the experiments, the size of a UDP datagram was 1.47 KB.

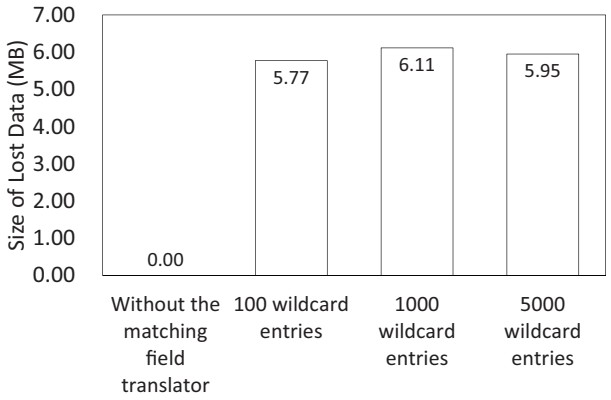


Figure 9. Lost data size vs. the wildcard entries in the flow entry translator.

B. Results

We measured the total lost data size for the first 1 seconds of sending UDP packets. Note that, regardless of the number of the wildcard flow entries, packet loss ended within 1 second in our experiment. Figure 9 shows the sizes of the lost data based on the numbers of wildcard entries in the flow entry translator and also for the cases where the OpenFlow controller was connected directly to the switch. The average for 10 experiments is shown.

As expected, a certain amount of data was lost when the flow entry translator was used. In this experiment, the size of the lost data was in the range of 5.77–6.11 MB. At most, the data for $1.47\text{KB} * 544\text{packets} = 799.68\text{KB}$ was buffered in the switch. Note that it was not possible to measure the precise size of the buffered data because the buffer was in the shared memory in the switch. The sum of the lost data and the rough estimate for the size of the buffered data is less than 7 MB. This is comparable with the buffer sizes in current top-of-rack (ToR) switches, e.g., the HP 5900 has 9MB of buffer space [9]. In general, the necessary buffer size depends on the throughput and the rate at which data packets with new header fields are arriving. Larger buffer sizes may be necessary. However, since buffers can be implemented using SRAM, we believe that the technique that we have proposed is capable of enabling low cost switches for the OpenFlow infrastructure. The cost is lower than the cost of using TCAM and complex chipsets.

VI. Related Work

Other than the approach that is described in Section II for reducing TCAM power, there are also approaches that reduce the cost of OpenFlow switches.

A. Software switch implementation

Software-based OpenFlow switch implementations uses general purpose processors (GPP) and DRAM or SRAM devices. The software implementation reduces the CAPEX and the OPEX of the infrastructure significantly. Open vSwitch [8] is the most widely used software for OpenFlow switch functionality on Linux systems. Dedicated hardware is not required when Open vSwitch is being used. It is designed for edge switches for virtual machines in data centers. Matsumoto et

 ISBN: 978-1-61208-318-6

al. [10] proposed a hash searching method for high speed matching to flow entries in RAM. For the exact matching fields for a flow entry, the hash value is calculated immediately when it is installed. For the wildcard matching fields for a flow entry, matching for the packet that arrives first is performed using one-by-one comparisons to flow entries in RAM. Then, the hash value of the header in the first data packet is calculated and the value is used for matching the subsequent data packets. A software switch is not suitable for forwarding the massive volumes of data packets in carrier access networks.

B. Product OpenFlow switches

Hybrid OpenFlow switches from NEC, HP, IBM, and Juniper use combination of general purpose and dedicated hardware devices. There are various patterns of combinations. Examples include combinations of application-specific integrated circuits (ASICs) and SRAM, combinations of TCAM for a limited number flow entries and SRAM for other flow entries, etc. The hybrid implementations can obtain high performance levels and reduce costs slightly. We propose a technique that would dramatically reduce the complexity of an OpenFlow switch implementation and also achieve line-rate packet forwarding performance levels.

C. IP source routing-like approach

Source flow [11] proposes an IP source routing-like approach. This technique embeds the action lists for data packets from intermediates switches in the network into the header fields at the ingress switches. Switches inside the network base their actions on the pointers in the header fields. As a result, the number of entries in the switch is the same as the number of actions in the switch. Because the number of flow entries is typically lower than of the number of actions that have been performed in the switch, the technique can reduce the number of entries in switches. The action list for the switch is retrieved from a centralized controller like an OpenFlow controller. The constraint on this approach is that all of the actions have to be supplied for the data packet at the ingress switch at once. Our proposal, on the other hand, virtualizes an OpenFlow network completely, i.e., an OpenFlow controller can see flows by obtaining the flow statistics at every hop in the network. As a result, the proposed functionality offers a higher level of control of the network than the source flow technique.

D. Label switching

The label switching techniques enable all switches in the network to match using only a single header that is referred to as the label. At the ingress switch of the network, the label for the data packet is added to the header fields of the data packet or the header fields are replaced using the value from the label. Inside the network, switches do not need to have TCAM. Furthermore, the length of the matching field is short. As a result, less BCAM space is sufficient. MPLS [12] is widely used in carrier core networks. The MPLS label is added at the ingress switch based on the destination IP address. In the network, switches forward data packets based simply on the label. However, MPLS does not have the same level of programmability as OpenFlow because it has two constraints: inflexible flow definitions (i.e., the label is based on the destination IP address) and a lack of a global view of the network.

PortLand [13] is designed to control routing for data center communications using the MAC header as the label in order to support the communications for many hosts at different sites using a large space for a MAC addresses. For every data packet that is destined for another site in the data center, the MAC header is replaced at the gateway. The new MAC address corresponds to the original MAC address and the ID of the destination site. This correspondence is managed globally. The MAC address is recovered at the gateway of the destination site. The difference between PortLand and our proposal is that PortLand is not designed to virtualize an OpenFlow network, but is designed for construction of a large-scale L2 network without the ARP broadcast overhead. On the other hand, our proposal supports arbitrary flow definitions in the network.

VII. Conclusions and Future Work

This paper proposed a technique that enables construction of OpenFlow networks using switches that have little more than L2 switch functionality. Arbitrary flow definitions from an OpenFlow controller are translated into flow definitions that are based on the MAC ID address at the external matching field translator and the flow entries are installed into the switch. For the wildcard matching fields in the flow entry, the corresponding exact matching fields are determined and the flow entry is installed into the switch after the first arrival of the matching data packet in order to avoid using TCAM. For translations in the data plane, the matching field translator manages the edge switches in order to modify the MAC headers of data packets. In our proposal, the OpenFlow interface is required in the switches inside the network in order to manage the flow table externally and enable simply matches on the MAC header. We believe that this type of switch can be assembled as an extension of an L2 switch.

A future work will focus on a distributed implementation for the matching field translator for scalability in carrier access networks. In the architecture of the proposed technique, the single matching field translator manages all of the switches in the network. The functions of the packet header and the matching field translators involve independent processes for individual switches in the network. As a result, they can be implemented easily in a distributed manner. Furthermore, the global correspondence of the original header fields and the MAC ID address can be managed easily by a distributed lookup system, e.g., a distributed hash table. However, we need to explore fast sharing schemes for the correspondence data because the distributed version of the matching field translator will be deployed at geographically remote sites. The time that is required in order to reference the correspondence is critical for the performance of the packet header and the matching field translators.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, et al., "OpenFlow: Enabling innovation in campus networks," in Proceedings of the ACM SIGCOMM 2008 Conference, vol. 38, no. 2, 2008, pp. 69–74.

[2] "SPARC deliverable d2.1: Initial definition of use cases and carrier requirements," 2010, [retrieved: Dec. 2013]. [Online]. Available: http://www.fp7-sparc.eu/home/deliverables/

 ISBN: 978-1-61208-318-6

[3] K. Pagiamtzis and A. Sheikholeslami, “Content-addressable memory (CAM) circuits and architectures: A tutorial and survey,” IEEE Journal of Solid-State Circuits, vol. 41, no. 3, 2006, pp. 712–727.

[4] J. Liao, “SDN system performance,” 2012, [retrieved: Dec. 2013]. [Online]. Available: http://pica8.org/blogs/?p=201

[5] “OpenFlow switch specification version 1.3.0 (wired protocol 0x04),” 2012, [retrieved: Dec. 2013]. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf

[6] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee, “DevoFlow: Scaling flow management for high-performance networks,” in Proceedings of the ACM SIGCOMM 2011 Conference, 2011, pp. 254–265.

[7] P. Congdon, P. Mohapatra, M. Farrens, and V. Akella, “Simultaneously reducing latency and power consumption in openflow switches,” IEEE/ACM Transactions on Networking, 2013, to appear.

[8] “Open vSwitch,” [retrieved: Dec. 2013]. [Online]. Available: http://openvswitch.org/

[9] “QuickSpecs HP 5900 Switch Series,” [retrieved: Dec. 2013]. [Online]. Available: http://h18004.www1.hp.com/products/quickspecs/14252_div/14252_div.pdf

[10] N. Matsumoto and M. Hayashi, “Performance improvements of flow switching with automatic maintainance of hash table assisted by wild-card flow entries,” in Proceedings of the 10th International Conference on Optical Internet (COIN 2012), 2012.

[11] Y. Chiba, Y. Shinohara, and H. Shimonishi, “Source flow: handling millions of flows on flow-based nodes,” in Proceedings of the ACM SIGCOMM 2010 conference, 2010, pp. 465–466.

[12] “RFC 2031: Multiprotocol label switching architecture,” 2001, [retrieved: Dec. 2013]. [Online]. Available: http://tools.ietf.org/html/rfc3031

[13] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, et al., “PortLand: A scalable fault-tolerant layer 2 data center network fabric,” in Proceedings of ACM SIGCOMM 2009 Conference, 2009, pp. 39–50.

 ISBN: 978-1-61208-318-6

DCPortalsNg: Efficient Isolation of Tenant Networks in Virtualized Datacenters

Heitor M. B. Moraes, Rogério V. Nunes, and Dorgival Guedes
Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, MG – Brazil
Email: {motta,rogervn,dorgival}@dcc.ufmg.br

***Abstract*—Multi-tenant datacenters have become an important scenario in this age of Cloud Computing. One important element for their effective deployment is the isolation of traffic from each tenant within the datacenter. In this work, we present *DCPortalsNg*, a solution based on the Software-Defined Network (SDN) approach, which provide effective and scalable isolation for virtualized datacenters. By adopting a per-tenant virtual network description, we can use an SDN controller to rewrite packets that flow through the physical network. That way, we can easily control which virtual machines they can reach. In our implementation, we leverage OpenStack Neutron's network representation to achieve a simple and extensible solution. Experiments show good results with little overhead, even preventing DoS attacks between tenants.**

Keywords*—*Software Defined Networks; virtual networks; OpenFlow.

I. Introduction

With the large adoption of Cloud-based solutions, multi-tenant, virtualized datacenters have become an important deployment scenario. In them, each user (the tenant) uses the datacenter hardware to host a set of virtual machines, configured to provide a specific service to his clients. One important element in that architecture is the proper isolation of traffic between tenants. That is necessary to guarantee the privacy and safety of each tenant's data, as well as to avoid unexpected traffic (malicious or not) to hurt the performance of any application in the datacenter. While machine virtualization provides good CPU, memory and storage isolation, there is still a need for better network virtualization solutions, specially when scalability and easy management are also expected [1].

Some datacenters rely on the use of VLANs to isolate each tenant's traffic. Although relatively simple, VLANs are limited by the protocol, and in some cases that may hinder scalability [2]. On such solutions, there is still the problem of handling the addresses of the large number of VMs in a single datacenter network. In some cases, each physical machine can host on the order of hundreds of VMs, each with its own layer 2 (MAC) address. Forwarding tables in most Ethernet switches have limited space, and performance can drop significantly if they cannot hold all addresses observed. Solutions based on protocol layering (tunneling) can reduce the need for switches in the network to learn all addresses, but they have performance and management limitations. A good review of these techniques is the work of Cabuk *et al.* [2].

With the advent of Software Defined Networks (SDN), it has become possible to program the network, so that new functionalities and behaviors can be added to switches in the network. Some new protocols and network architectures are being proposed to address those issues, but they require new hardware to operate [3], [4]. In most cases, upgrading all the network hardware is not an option, and scalable, cost-effective, easy-to-manage solutions are still missing. In a previous work [5], we proposed a software-only solution using an SDN controller and the virtual switches at the edge of the (virtualized) datacenter network. Although effective, that solution had some major limitations, primarily in the way tenants could represent their virtual networks and in the way it achieved isolation, which limited the tenants hability to use any IP addresses they chose for their networks (specially restricted addresses).

In this paper, we describe *DCPortalsNg*, which addresses those issues without requiring new hardware. We make use of the new OpenStack Neutron component (http://www.openstack.org/) to handle the tenant networks and therefore create a more detailed and flexible (yet simpler) representation. With that representation, a better packet rewriting scheme can be applied that, at the same time, gives tenants more flexibility to define their address schemes, and simplifies the processing of packets as they cross the network. By doing that, we hide real traffic origins and destinations from the core of the network (hardware), also hiding traffic from each virtual network from any VMs not belonging to the same tenant.

The abstraction of the SDN network hypervisor provides a logically centralized location where network configuration and control can be performed easily, while maintaining the scalability of the solution. There are important benefits to this approach, like *(i)* a reduced demand for the conventional switches' forwarding memory, since VM addresses are hidden from core switches, *(ii)* the creation of isolated virtual networks for each tenant, guaranteeing that one tenant's traffic will never reach VMs of others, *(iii)* integration of VM and virtual network configuration and management, by integrating the network hypervisor with OpenStack, *(iv)* it leaves VLAN tags free to be used for other purposes, such as implementing VLAN-based multi-path routing, for example [6].

With that in mind, the remainder of this paper is organized as follows: Section II describes the architecture and operation

 ISBN: 978-1-61208-318-6

of the system, while its behavior is evaluated in Section III. After that, Section IV puts *DCPortalsNg* in context, describing related work, and Section V provides some concluding remarks, as well as some observations about future work.

II. Implementation

DCPortalsNg was implemented as a network hypervisor module built on top of the POX SDN controller (http://www.noxrepo.org/pox/about-pox/). It interfaces with *OpenStack* through a Neutron plugin, which provides the information it needs about virtual machines and their virtual networks, such as tenant identification, other VMs in a given network and, specially, VM location. With that information, it builds OpenFlow messages to tell the Open vSwitches how to handle packet flows from/to a given VM. OpenStack controls the hypervisor in each host, which configures its Open vSwitch accordingly. Figure 1 illustrates relations between the modules.

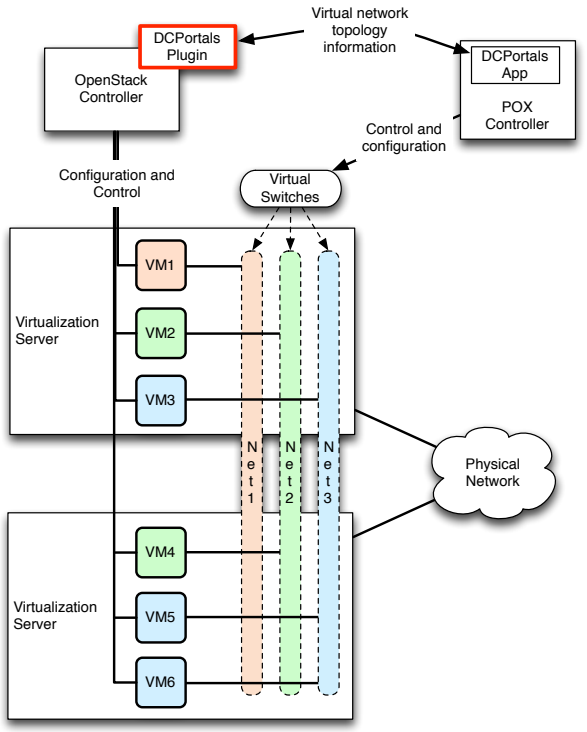


Fig. 1. Architecture of the *DCPortalsNg*.

The system administrator uses the OpenStack API to describe the virtual network for each tenant and to start each virtual machine which belongs to it. OpenStack selects the physical machine that will host the VM and sends the information it needs to run. Through the Neutron plugin, it informs *DCPortalsNg* about the network topology and the VM connections. The hypervisor connects that VM to the Open vSwitch of that physical host. POX can, then, assign a unique ID for each VM located in a certain host (VM IDs are unique for the whole network, but are kept organized by host machine). That information is cached in memory for quick access, when needed.

When the new VM sends its first packet through the network, the Open vSwitch identifies a new flow and uses the OpenFlow protocol to inform POX. It then notifies *DCPortalsNg*, which handles the packet, accessing its data structures to recover the information about the source and destination VMs. If the two machines are not in the same network the flow is not allowed and the packet is dropped. Otherwise, the system commands POX to send another OpenFlow message back to the appropriate Open vSwitch, telling it how to handle all future packets between those two VMs. In *DCPortalsNg*, a flow is identified by origin/destination MAC addresses only, so it will apply to all traffic between the same VMs. The sections that follow provide more details about this process and each of the main aspects of the system.

A. The virtual network abstraction

DCPortalsNg takes advantage of OpenStack Neutron to derive its information about each tenant's virtual network topology. Neutron manages three kinds of entities for each tenant: the network, which can be seen as a switch connecting all the VMs from that tenant, the subnet, which defines the address range for the network, and the ports, that represent the connection of each VM to the network. With that representation, each tenant determines which machines are allowed to communicate to each other (those connected to the same networks) or not. That abstraction guarantees at least the security of a local network, without forcing the tenant to worry about the physical, shared, infrastructure.

Our system builds a set of directories to hold that information: one mapping networks to tenants and *vice-versa*, one to hold the info about ports (VMs) associated with each network, and finally one to map VMs to physical hosts. In all cases, VMs are represented by unique identifiers created by the system as addresses in a 10.0.0.0/8 address space (the same principle could be used with IPv6, which is supported by OpenFlow since version 1.3). Although represented as an IP address, that identifier has no direct relation to the VMs' real IP addresses. With that representation, given a VM identifier, *DCPortalsNg* can recover its network, MAC and IP addresses, physical host and even the Open vSwitch port to which it is connected. That information will be used during the decision process needed to route packets.

B. Packet rewriting for network isolation

As previously discussed, using packet rewriting to implement network isolation has two major benefits: it guarantees that traffic from a tenant will be out of reach for others, and it reduces the pressure on physical network devices to handle MAC addresses for all virtual machines in the datacenter. To achieve that, we rewrite the MAC addresses in all packets that traverse an Open vSwitch at the edge of the network to remove the VM information.

For now, we can assume that *DCPortalsNg* has already identified the associated flow and programmed the edge switches accordingly to rewrite the packet before forwarding it. The process described next is illustrated by Figure 2. In it, virtual

 ISBN: 978-1-61208-318-6

machine *vm1*, operating in physical host machine *host1*, sends a packet to another virtual machine *vm3* in the same virtual network, but physically located in physical host machine *host2*. The original packet sent by *vm1* that will reach the virtual switch at *host1* will contain the MAC addresses of *vm1* and *vm3*, and the IP addresses of both.

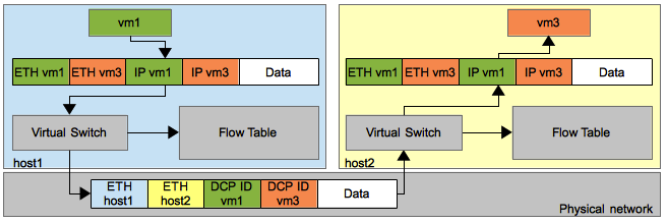


Fig. 2. Address rewriting in *DCPortalsNg*

The Open vSwitch at *host1*, then, will replace *vm1*'s and *vm3*'s MAC addresses in the Ethernet packet header with the MAC addresses of *host1* and *host2*. At the same time, it will replace their IP addresses by their *DCPortalsNg* internal identifiers. Routing at the core Ethernet network will be done in terms of the physical machines' MAC addresses. When the packet reaches *host2*, it will traverse its virtual switch; at that moment, an OpenFlow rule already set in place by *DCPortalsNg*, based on the IDs of the two VMs, will write back the appropriate MAC and IP addresses corresponding to *vm1* and *vm3* in the header. That is the packet that will be delivered to *vm3* at that point. Notice that the MAC addresses of *vm1* and *vm3* never crossed the network; nevertheless, they will only reach their destinations if the system can verify their connectivity to a same virtual network.

C. ARP messages

The MAC rewrite technique just discussed replaces the Ethernet addresses of virtual machines in packets that were already built with those addresses. However, for the VMs to build those packets in the first place, they must learn the MAC address of the destination. In a traditional network, that would be achieved by a broadcast message using the Address Resolution Protocol (ARP). The protocol is composed basically by two kinds of messages: ARP Request and ARP Reply. The first one is sent to the network broadcast address when we need to learn the MAC address associated with a certain IP address. The second is the response, sent by the target machine, to inform the sender of the query about its MAC address.

For the virtual network isolation to work, it is not acceptable that broadcast messages travel the network carrying virtual machine MAC addresses. *DCPortalsNg* fixes this by intercepting all ARP communication and handling it directly. Since it has access to the OpenStack database, it knows how to answer to any ARP query in the network. All it has to do in this case is to build an ARP Reply message with the right information and send it directly back to the appropriate host. Since queries are intercepted at the virtual switch close to the sender, ARP messages never cross the core network.

D. Inter-networking

In modern datacenters, there are cases where two tenants may allow their applications to communicate with one another (based on some mutual agreement). There are also many cases where a tenant's machines must be accessible by clients from the Internet. To achieve that, networks described by the tenants may include mentions to special routers.

The details of how this connection is defined depend heavily on each datacenter structure and service policy. It might be offered only through the definition of a second interface in one of the virtual machines, which would be the only one visible externally, while that host would be responsible to route messages between the internal, virtual network and the outside network.

Although these might be implemented as actual multi-homed VMs, the SDN approach allows us to simplify that, avoiding the need for extra virtual machines: *DCPortalsNg* can simply add rules to directly rewrite packets from the origin network to the destination network, by using special MAC addresses to identify the (abstract) routers.

E. Broadcasts

Although most broadcasts in local networks are ARP-related and, therefore, eliminated by *DCPortalsNg*, we must still consider how other broadcasts are to be handled. When a group of VMs is configured in a virtual network, we expect packets sent to that network's broadcast address to be delivered to all machines in that virtual network, and only to them. However, in a complex, shared environment like current datacenters, that is not the case, since packets would be delivered to all machines connected to the physical Ethernet network.

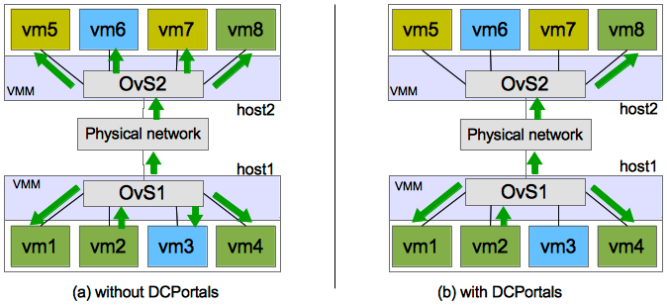


Fig. 3. Difference of treatment of a broadcast

Figure 3 shows the difference between what happens in that case with and without *DCPortalsNg* in the network. In the figure, virtual machines are separated in different virtual networks according to their colors and *vm2* sends a broadcast message. Ideally, that message should be delivered only to the other virtual machines in the same network, *vm1*, *vm4*, and *vm8*. Without *DCPortalsNg*, however, in the conventional system, all machines, no matter to what virtual network they belong, would receive the packet.

 ISBN: 978-1-61208-318-6

To achieve the desired effect, first the broadcast packet is inspected by the virtual switch at the physical machine where the sender VM runs. It is then delivered to the ports of the local switch that are connected to other VMs in the same network, and then *DCPortalsNg* must make sure it is received by all other VMs of that tenant located in other hosts.

The transmission to reach other hosts may be done either by an Ethernet broadcast or by packets addressed to the physical interface of each host holding VMs from that network. One solution has lower latency, while the other will avoid broadcasts flooding a large network.

In any case, when the packet leaves the host machine of the sender, the process of MAC rewriting works as before, to remove the MAC address of the sender, but keeps the Ethernet broadcast address as destination (in the first approach). When the broadcast message reaches the virtual switches at the destination physical machines, in both cases, the controller restores the original headers and also searches for all local virtual machines that belong to the network of the sender, delivering the message to each of them, and only to them. In our example, the controller would program the switch OvS2 to deliver the packet only through the port connected to vm8. By doing that, the message continues to have the effect of a broadcast, but it will only be delivered to machines in the same virtual network of the sender, guaranteeing no other machine will have access to the packet.

For the evaluation tests described later, *DCPortalsNg* maintained the message as a broadcast in the physical network, since that would stress the network further.

III. Evaluation

To perform the validation experiments to confirm the proper operation of the system, we used three machines, each with two network interfaces, connected to two different switches. One of the resulting networks was used for management traffic (OpenStack commands, OpenFlow), and the other was used for the communication between the VMs (operational network). Such configuration is very common in commercial datacenters [1].

The management network uses IP addresses in the range 10.0.254/24. Virtual machines were configured in two separate virtual networks, each spanning two physical hosts. (A single address range was used for all VMs, to stress the need for traffic isolation: being configured with IP addresses in the same range, unless some external isolation is active, traffic from a host can reach all the other hosts.) Two virtual networks were created and machines were distributed according to the configuration described in Table I.

Host machines were configured with the Ubuntu Operating System, version 11.10, and packages *OpenStack*, *Xen*, and *Open vSwitch* from official repositories. The virtual machines were configured with one of OpenStack's standard images, running Ubuntu 10.10.

The first experiment was a simple isolation test using ping to the network broadcast address and the network latency. The second one evaluated the interference of the system on communication latency, also using ping. Finally, the last one tested the system under a denial-of-service condition.

TABLE I
Distribution of virtual machines and their networks among the physical machines for the evaluation tests.

Virtual Machine	IP	Virtual Net	Host
vm2	10.0.20.2	lan1	host1
vm3	10.0.20.3		
vm4	10.0.20.4		host2
vm5	10.0.20.5	lan2	
vm6	10.0.20.6		

A. Isolation and latency overhead

The confirmation of isolation consisted in using ping to send an ICMP Request messages from one of the virtual machines to the network broadcast address (the operating system of the VMs was configured to enable ICMP replies to broadcast requests). The expected behavior for this use of ping is that all machines in the same network as the sender should reply to the sender. When using *DCPortalsNg*, even with all machines using the same IP address range, with the same broadcast address, and sharing the same infrastructure, only virtual machines in the same virtual network as the sender should receive the request and, therefore, reply to it. That was the observed behavior in each case, confirming the proper isolation (program output removed due to space limitations).

We also collected packet traces at the interfaces of the virtual machines (therefore, inside their virtual networks) and at the network interface of the physical hosts (at the point where a packet enters the network core, past the virtual edge switches). By inspecting those traces, we confirmed that packet rewriting occurred as expected.

To verify general latency overheads, we also used ping, now between two VMs in a same virtual network. We discarded the first packet, to discard the ARP and OpenFlow setup overheads (discussed next). We repeated the pings 10,000 times for each scenario and computed a 99% confidence interval for the results, which showed that *DCPortalsNg* overhead in this case was approximately 1% in the worst case.

B. Network setup overhead

It is important to quantify the impact *DCPortalsNg* may have on network latency for a working flow, considering the configuration setup activity. For this experiment, we again used ping messages. We considered three scenarios: first, we configured all Open vSwitch instances to operate as standard Ethernet switches — the default setup, which is used in traditional network configurations. That gives us our baseline case. Next, we configured the Open vSwitches as OpenFlow switches, but used a streamlined network hypervisor that just emulated the operation of an Ethernet learning switch. That would show us the overhead of just having OpenFlow active, without adding the costs of our system. Finally, we considered a complete *DCPortalsNg* installation, where all those costs were included.

 ISBN: 978-1-61208-318-6

First, we evaluated the overhead for the first packet of a flow, when the sequence of actions differs the most between a traditional network and an SDN. As discussed earlier, when that happens in an SDN, the packet is forwarded to the network hypervisor, which must decide what will be done with that flow and send a command back to the switch. In *DCPortalsNg*, that will include executing queries to OpenStack to retrieve information for each VM involved. That may also include an ARP query. We considered both the cases when that query is necessary and when the ARP table at the machines already contains the MAC addresses for the endpoints involved in the communication. When measuring ARP overhead, we isolated its processing by POX by pre-programming the switches forwarding tables, so the network hypervisor would be contacted only to process the ARP request, not for the actual flow. On the other hand, when measuring the flow setup overhead, we pre-programmed the ARP tables, so that no ARP queries were issued. As a baseline, we also measured times for a standard system, where Open vSwitches were configured as standard Ethernet switches, with no OpenFlow. For each scenario, we ran 30 pings and recorded the round-trip times observed. We also computed the statistical difference between each pair of scenarios. Table II shows the average results, with errors for a confidence interval of 99%.

TABLE II
SETUP OVERHEAD FOR EACH SCENARIO, FOR BOTH ISSUING AN ARP QUERY AND INSTALLING A FORWARDING RULE AT THE EDGE SWITCHES. THE STANDARD SWITCH CASE IS SHOWN AS A BASELINE.

Scenario	ARP overhead (ms)	Flow setup (ms)
Standard switch	7.45 +/– 0.1	0.20 +/– 0.03
POX L2 switch	10.72 +/– 2.67	29.83 +/– 8.36
DCPortalsNg	15.31 +/– 3.57	47.05 +/– 8.96

We see that ARP costs vary less than those of flow setup. That is due to the fact that in a standard scenario, ARP requires a broadcast that will reach the destination and a message back; for the POX L2 switch, there is still a network hypervisor involved, but all it does is to return the packets to the switch for delivery as it would be done in the standard switch; finally, for *DCPortalsNg*, an ARP query is transformed into a message to the network hypervisor, which searches an internal table for proper info, and a reply is sent directly to the original sender (there is no contact to the destination machine).

Flow setup costs, however, have a higher variance. There is basically no setup cost for a standard switch; the time shown, 0.20 ms, is just the ping round-trip time through the network. The POX L2 switch must contact the network hypervisor, which will reply by installing a flow based on the addresses it learns during the process, so we can consider that the OpenFlow processing overhead. *DCPortalsNg* adds to that the cost of querying its dictionaries to identify the endpoints and set up the forwarding table. Although there is a significant setup overhead in this case, it is important to remember that it only takes place at the beginning of the communication between two VMs when a flow is set. Besides that, the overhead is less than 50 ms, which would not trigger retransmissions in a TCP connection.

C. Denial of service attack protection

One common motivation for virtual network isolation is the threat that a tenant may start a denial of service attack targeted at another tenant's machines. In a datacenter environment where there is no such isolation, a UDP flow created from an attacking machine to the target network may drain network bandwidth to a point were the attacked system cease to function. One similar attack happened to the BitBucket service, while using Amazon EC2 infrastructure [7] (although, in that case, the UDP traffic came from outside the datacenter).

Such a problem should not happen if the tenants' virtual networks were properly isolated from each other and from the outside. To verify that, we created a UDP flow attack to another virtual network. To make things worse, the UDP flow was created with the broadcast address of the target network as destination. Figure 4 shows the experiment setup in this case.

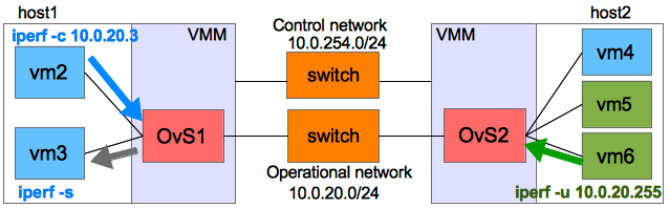


Fig. 4. Denial-of-service experiment. VM6 launches a UDP flood to the network broadcast address, while VM2 and VM3 (in another virtual network) set up a TCP flow.

Virtual machines vm2 and vm3, located at physical host 1, set up a TCP transfer between them. At the same time, vm5, the attacker, initiated a UDP flow addressed to the other virtual network's broadcast address. The attack to the broadcast address is a worst case scenario, since it would be processed by both vm2 and vm3, if delivered. If the attack was addressed to any machine's IP address only, the effect with no control might be slightly less damaging, but the behavior with *DCPortalsNg* would be the same. We used iperf to create the two flows, and measured the effective throughput of the TCP connection between vm2 and vm3. We limited the bandwidth of each virtual machine to 1 Gbps, a common value in practice.

Each test run lasted 1,000 seconds and the TCP throughput was measured every 3 seconds. The first 3 seconds were discarded to eliminate setup variations. Results are shown in Table III.

TABLE III
AVERAGE TCP THROUGHPUT OBSERVED IN DIFFERENT CONDITIONS; RANGES CONSIDER A 99% CONFIDENCE INTERVAL.

Scenario	Bandwidth (Mbps)
No isolation, no attack	928.29 +/– 0.16
DCPortalsNg, no attack	928.21 +/– 0.22
No isolation, under attack	41.21 +/– 12.99
DCPortalsNg, under attack	910.11 +/– 0.28

 ISBN: 978-1-61208-318-6

Clearly, we can see the difference in the two cases. The system with no isolation suffers a loss of about 95% of the observed throughput. *DCPortalsNg* suffers just about 5% loss. With isolation, the UDP flow is blocked at the edge switches, not being delivered to other networks. The loss in this case is due to the overhead at the edge switch to drop the incoming UDP packets. Considering the system with no attacks, there is no statistical difference between *DCPortalsNg* and the standard switch scenario. That should be expected, considering the previous analysis of latency overheads.

IV. Related Work

As we mentioned in the introduction, this work is a continuation of our previous work in the area [5]. Compared to that work, *DCPortalsNg* provides a better integration with OpenStack, a simpler API to define the tenant networks, and an improved rewriting scheme.

Greenberg *et al.* [1] put together an interesting study about costs in a cloud computing datacenter. Among other observations, the network is identified as one of the major challenges in that context. Authors explicitly mention the dependence of current solutions on VLANs, and the problems associated with that practice.

One of the first initiatives towards a technique for virtual networks isolation was developed by a group at HP Labs [8], beginning with the definition of the concept of Trusted Virtual Domains (TVDs). Those would be logically isolated network sections, independent of the infrastructure topology. To implement that isolation, the authors implement a module internal to the virtual machines that is responsible for all processing related to network isolation. Two techniques for isolation, VLAN tagging and the EtherIP encapsulation, are compared. That work has a similar goal to *DCPortalsNg*, but the the solutions considered have scalability limitations and require intrusive modifications to the hypervisor. A longer comparative study by the same group mentions the MAC rewriting technique [2].

DCPortalsNg uses the SDN paradigm to solve the isolation problem. Pettit *et al.* [9] have already discussed the viability of such approach to datacenter networks, but did not present a concrete solution. Two applications of NOX (another SDN controller) to the datacenter were published previously, but they focused on implementing new network architectures and traffic control [3], [4]. Different from those solutions, *DCPortalsNg* does not require hardware with OpenFlow capabilities inside the network core, and focuses only on traffic isolation.

One work with very similar motivation to the one presented here is certainly Netlord, developed by Mudigonda *et. al.* [10]. In their paper, the authors use a solution based on encapsulation to achieve a similar traffic isolation without requiring special hardware in the network. However, the way their solution is implemented is quite different, using an extension of the Xen hypervisor specially developed for that goal. We believe that the use of Software Defined Networks is a more elegant and flexible approach and a determinant characteristic of our work. It simplifies implementation and offers a more flexible solution. *DCPortalsNg*, for example, works directly not only with Xen but with other hypervisors that use libvirt and Open vSwitch, such as KVM.

V. Conclusion and Future Work

This work presented *DCPortalsNg*, a system developed to provide traffic isolation for virtual networks in a virtualized datacenter environment. The system architecture and the implementation details were described, along with results that confirm the isolation provided. Evaluations also quantified the overheads during flow setup, which are noticeable but rare, and showed that during normal flow operational costs are negligible. Finally, we showed that the system can be effective at protecting tenants from denial-of-server attacks inside the datacenter network. As future work, we continue to improve the system. In particular, we are working on integrating *DCPortalsNg*, which provides network isolation, with Gatekeeper, a system designed to provide network traffic guarantees in a datacenter environment [11].

Acknowledgments

This work was partially sponsored by UOL, Fapemig, CNPq, and the National Institute of Science and Technology of the Web, InWeb (MCT/CNPq 573871/2008-6).

References

[1] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2009.

[2] S. Cabuk, C. I. Dalton, A. Eduards, and A. Fischer, "A comparative study on secure network virtualization," HP Laboratories, Tech. Rep. HPL-2008-57, 2008.

[3] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker, "Applying NOX to the datacenter," in *Proceedings of workshop on Hot Topics in Networks (HotNets-VIII)*, 2009, pp. 1–6.

[4] B. Heller *et al.*, "Ripcord: a modular platform for data center networking," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 457–458, 2010.

[5] R. V. Nunes, R. L. Pontes, and D. Guedes, "Virtualized network isolation using software defined networks," in *Proceedings of the 38th IEEE Conference on Local Computer Networks (LCN)*. IEEE, 2013, pp. 700–703.

[6] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul, "Spain: Cots data-center ethernet for multipathing over arbitrary topologies," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, ser. NSDI'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–16.

[7] "Bitbucket amazon ddos attack," http://blog.bitbucket.org/2009/10/04/on-our-extended-downtime-amazon-and-whats-coming/ (retrieved: Dec 2013), 2012.

[8] S. Cabuk, C. I. Dalton, H. Ramasamy, and M. Schunter, "Towards automated provisioning of secure virtualized networks," in *Proceedings of the 14th ACM conference on Computer and communications security*, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 235–245.

[9] J. Pettit, J. Gross, B. Pfaff, M. Casado, and S. Crosby, "Virtual switching in an era of advanced edges," in *Proceedings of the 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC CAVES)*, ser. DC CAVES. Amsterdam: ITC, september 2010, pp. 1–7.

[10] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary, "Netlord: a scalable multi-tenant network architecture for virtualized datacenters," in *Proceedings of the ACM SIGCOMM 2011 conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 62–73.

[11] H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes, "Gatekeeper: supporting bandwidth guarantees for multi-tenant datacenter networks," in *Proceedings of the 3rd conference on I/O virtualization*, ser. WIOV'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6.

 ISBN: 978-1-61208-318-6

Heterogeneous Virtual Intelligent Transport Systems and Services in Cloud Environments

Vladimir Zaborovsky, Vladimir Muliukha, Sergey Popov, Alexey Lukashin
Telematics department, St. Petersburg State Polytechnical University
Saint-Petersburg, Russia
e-mail: vlad@neva.ru, vladimir@mail.neva.ru, popovserge@spbstu.ru, lukash@neva.ru

***Abstract* — Modern Intelligent Transport Systems (ITS) are based on specific services that are hosted at network edges. These services are created using in-vehicle computing appliances, private cloud infrastructure and global information resources. According to the proposed approach all vehicles are considered as a mobile part of a low level operation network that provides low latency and requested quality of service (QoS) characteristics for the intelligent transport communication and information systems. Moreover, some of the discussed decisions support multiprotocol interactions and provide predictable real time performance so they can be used for different kinds of industrial, transport and robotics applications. ITS supplemented by low level operation network expands opportunities for a practical implementation of the emerging technologies for the Internet of Things (IoT). Due to the wide functional abilities the proposed approach is suitable for Big Data and on-demand high performance applications. Some aspects of these services develop the ideas of IBM "Smarter Planet" initiative and CISCO' Fog Computing. Researched model of multiprotocol node may be seamlessly integrated into an existing ITS cloud infrastructure using virtual firewall appliances to provide bilateral access control between vehicles that belong to MESH network and IaaS segments' resources, which support high performance computing and even supercomputers services.**

Keywords – Intelligent Transport Systems; Cloud Services; MESH; Multiprotocol Node; Security Services

I. INTRODUCTION

There are few approaches with potential to improve reliability and security of Intelligent Transport Systems (ITS) using possibilities to merge local resources of vehicle and different kinds of global cloud oriented services. Some aspects of these services develop the ideas of IBM "Smarter Planet" [1] initiative and CISCO' Fog Computing [2]. Emergence is achieved by the integration of three key technologies into reconfigurable and scalable service infrastructure: information, computer and communications. Such combination process is not a trivial task, especially in the case of transport systems, which support interaction of moving objects. The advantage of this technological approach is to expand online services to the drivers and passengers, to increase logistics efficiency and security of transport operations, as well as to prevent some road accidents. In this context, main challenge concerning research and design for the new generation of ITS is associated with a collaborative decision of the fundamental problem – organization of a real time access to big data from a moving car [3]. All aspects of this problem have a common background that are closely associated with communication technology tasks, namely [4]:

Connecting a user to local and global data. The volume of data, to which the vehicle has an access, is a critical parameter of ITS. This requirement depends on a hierarchical structure of the territorially distributed communication system, the local part of which receives time-critical operational data and processes it by the end-users' computers, and the second one belongs to a global cloud-oriented distributed information service housing at the data center far from the edge of ITS infrastructure.

Distributing processing tasks between a vehicle and cloud backbone resources. A computing platform extends capabilities of ITS services by sharing a processing operation with data between mobile real time vehicles' appliances and high performance massive scale resources of global information network or private cloud backbone.

Support bilateral mobile vehicle interaction. Variability due to the mobility is the key feature of ITS, that should be taken into account to improve performance, security and privacy issues by controlling data flows at the networks' edge points and by integrating multiprotocol vehicles' gateway with distributed communication infrastructure.

Seamless integration with security services. Information security requires seamless integration of data and communication services. This can be reached by using specific solution based on the stealth firewall technology for vehicle telematics hardware appliances and IaaS components of cloud environment.

Taking into account all aspects mentioned above we consider new services for Vehicle Controls Systems (VCS) that are hosted at any edge of ITS network infrastructure. These services expand the range of automotive protocols supported by vehicle embedded computing appliances, as well as available via MESH network private cloud resources and global public information systems. From the system point of view new services can be divided into three main categories: 1) communication services, which support real-time requirements; 2) access control between vehicle and high performance data processing resources; 3) high capacity storage systems that are available to VCS and belong to ITS cloud environment. Discussed approach can be implemented not only within ITS but also for various applications including emergency departments, regional data centers and Internet of Things (IoT).

The paper is organized as follows: in Section II, we introduce basic requirements for a new generation of ITS, describe characteristics, architecture and data structure of new proposed services; then, in Section III, we propose a model of ITS network edge that is the key component of mobile transport MESH network; we move on with information

 ISBN: 978-1-61208-318-6

security and access control services based on stealth firewalls in Section IV and conclude with Section V, in which we briefly present the main results of our work.

II. ITS AND CLOUD TECHNOLOGY

One of the promising technologies for vehicles' infrastructure management is cloud computing [5]. This technology allows us to take into account the territorial distribution and dynamic nature of the transport systems while improving their sustainability and scalability. The synergistic effect of the cloud technology implementation include a number of advantages, namely:

- Improvement of the dynamic characteristic of the MESH network at physical and data links layers;
- Expansion of available information and computing services;
- Spreading end-user's requests between several access points and applications to reduce response time and to increase semantic significance of the responses.

The main technological challenge concerns the way of how to allocate widely spread services and provides their availability to the end-user. One of the perspective ways to solve the problems mentioned above is to use cloud-oriented approach, which can be extended for mobility application due to the delay tolerance and secure infrastructure services (see Fig. 1).

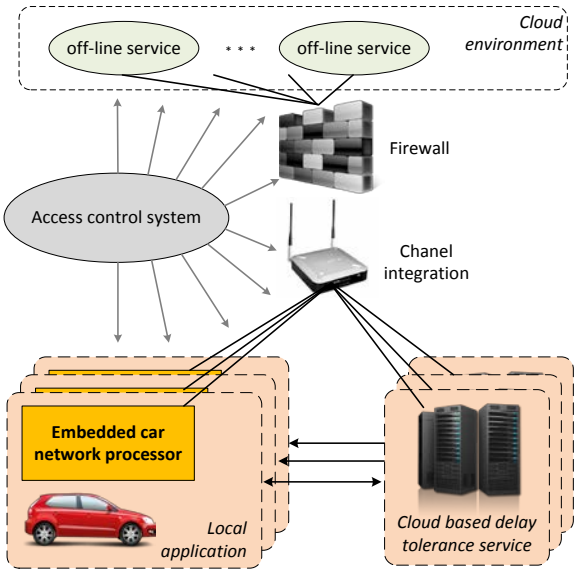


Figure 1. ITS system structure.

Each vehicle appears as a source of information in the cloud environment and for other vehicles simultaneously. The vehicle control system itself has a non-zero probability of crash or failure. However, data flows from vehicle via wireless media are susceptible to interferences that can disrupt connection and interaction of network notes and service components. In this case, the reasonable effectiveness criterion of the tasks listed above is the probability of the message delivery within a given time interval from the source to the destination in the cloud environment [6]. Delivery probability is a controlled parameter, which is a function of the cloud resources needed to process service requests. For example, fault tolerant implementation of supervisory tasks, such as route planning or vehicle speed control can be realized with the requested level of delivery probability by using the distributed cloud computing system. Furthermore, service agents, which interact with a vehicle applications and run on a remote Virtual Machine (VM) can improve or even optimize vehicle performance by analyzing historical driving patterns along the same route.

The concept of a virtual machine in the cloud architecture and MESH network architecture, that we propose, provides the implementation of the fault-tolerant, powerful and flexible tool for managing traffic services, roads infrastructure, and vehicle data. The implementation of our concept, which is based on the access control between vehicles' appliances and cloud services can be realized using existing wireless and wired connections of various technologies.

We consider a set of objects, a model of which is an extended network socket abstraction:

$$\mathrm{M^o} = \{\mathrm{name, IP, port}\}, \quad (1)$$

where *name* is a name of the service provided, *IP* is an address of the facility, and *port* is an applications' port, which is used for the transport layer interconnections.

On the set of objects M^o we introduce a number of services:

$$\mathrm{M^s} = \{\mathrm{name}, \{\langle \mathrm{p_i, type} \rangle\}\}, \quad (2)$$

where "*name*" is the name of the service, $\{p_i\}$ is a set of the typed parameters with the attributes type.

Consider the map $T = M^0 \circ M^S$, formed by the ratio $\Lambda(t)$ for different moments of time $t = t_0, \dots, t_n$, which determines the availability of the service for the subjects of the information exchange.

Then, oriented dynamic multigraph $G = (V, E)$, where V is a set of vertices, consisting of the named services from M^s, and E is a set of edges determining the sequence of services, characterizes the availability of the chosen sequence of services $\{m_1^s, \dots, m_n^s\} \in V$.

In turn, each vertex of the multigraph is a directed dynamic graph

$$\mathrm{g = (v, e)}, \quad (3)$$

where v is a set of service parameters of $\{p_i\}$, and e is a set of edges, that determine the acceptability of the given sequence of operations for the selected parameter $\{p_i\}$.

Within proposed hierarchy of models an admissibility of operations is characterized by quantifying estimation of relationships $\lambda_i(t)$, chosen from the set of all estimators of operations $\{\lambda\}$.

In this case, the task of choosing m^o from the set $\{M^o\}$ to obtain a sequence of services $\{m_1^s, \dots, m_n^s\}$ with parameters $\{\{p_1\}, \dots, \{p_n\}\}$ from $\{M^s\}$ can be formulated as the problem of finding a path of the dynamic multigraph G:

$$\exists\, (\mathrm{E_1 \dots E_n}) \in \mathrm{E}, \exists\, \{\mathrm{e_1, \dots, e_{1n}}\} \in \mathrm{e}\, |(\mathrm{E_1 \dots E_n}) = \\ = \{\mathrm{m_1^s \dots m_n^s}\} \vee \forall \mathrm{e_i} = \mathrm{p_i}\,. \quad (4)$$

Equation (4) can be solved by the modified Dijkstra's routing algorithm [3], in which for each moment is given the

vector of parameters, that takes into account both the information and the geographical connections of cyber-objects.

There are three ways to implement secure communications between vehicle and cloud services: vehicle-to-vehicle (V2V), vehicle-to-communication infrastructure (V2I), vehicle-to-cloud (V2C). For the first one, we need to use MESH topology [4], in the second case – multiprotocol telematics appliances, and the last way could be realized using reconfigurable wireless networks. Merger of all these technologies provide solid background for a bilateral information interaction between all parts of ITS. That leads to simultaneous use of various technologies for communication channels to improve accessibility of cloud services, which require integration of data communications to the shared wireless multiprotocol network.

Classical algorithms of MESH networks allow searching of the single route to the unique pre-known user. In the case of the cloud-oriented services and vehicles' appliances, while routing each time, a network node has to make a choice of the most perspective next hop from several alternatives. It is necessary to find available destination nodes with access to a cloud and to evaluate perspective of communication through them.

Fig. 2 shows the formation of functional virtual networks based on the multiprotocol MESH network of emergency services vehicles. Red background shows a virtual network of ambulance cars that provide an emergency aid service; blue one presents a virtual network of police cars.

Figure 2. Functional virtual networks of emergency services vehicles.

Local vehicles appliances communicate via MESH network with single vehicles (V2V component) and cloud services (V2I and V2C components) as they need. Communication with the cloud environment can be implemented in two ways: by the vehicle with communication equipment and by the stationary point. The vehicle located out of stationary communication area can access the cloud resources through a vehicle relay network. The network provides a bidirectional message transfer between the vehicle and the cloud environment.

The most important task for such network construction is the choice of communication protocols to increase message transfer adequacy. An estimation of message delivery time to the cloud services and back to MESH network users depends on a vehicle's traffic intensity, a level of communication network load, an interface availability and its composition in each vehicles.

A set of interfaces that allows being a user of Long Term Evolution (LTE) and MESH networks concurrently or MESH only (communication between vehicles only) should be installed in vehicles (network node). This new double-interface node (LTE, 802.11s) serves as a gateway providing communication between the MESH network and the cloud environment through LTE.

Data transfer protocol and an intensity of transfer determine message transfer adequacy, design and actual communication speed, mean latency of message delivery between vehicles' appliances and cloud environment.

The main features of the virtual communication network shown in the Fig. 2 are the following:

- Short lifetime of the static vehicles' MESH;
- A necessity of message transfer via MESH to the node that has an access to the cloud-oriented environment;
- Seamless integration with security services using cloud-oriented firewalls.

III. IMPLEMENTATION OF NEW MULTIPROTOCOL TECHNOLOGY IN FUTURE ITS

We need to realize multiprotocol support for message and data delivery within restricted time interval using different kinds of telecommunication protocols for fully use of cloud-based infrastructure to provide different vehicle's services, especially critical tasks. These aspects are the key requirements for future ITS that should operate with mobile objects and stationary components of infrastructure.

The implementation of such future ITS can be realized using network access mobile devices with reconfigurable multi-frequency radio connections that are simultaneously compatible with wireless interfaces (Fig. 3).

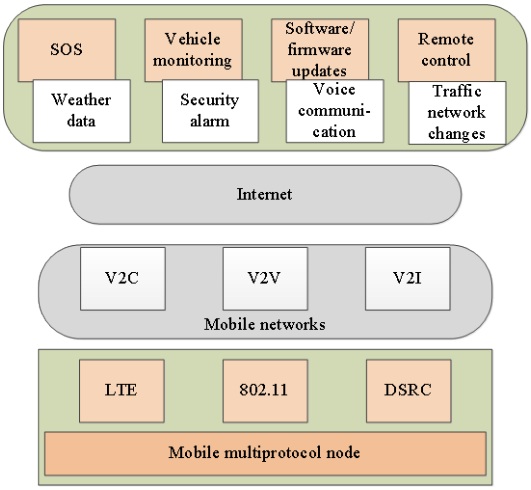


Figure 3. Multiprotocol node supporting ITS services.

The support of information interactions mentioned above requires designing a new generation of multi-protocol routers for Dedicated short-range communications (DSRC), LTE, MESH, and Wi-Fi networks. Such routers should generate optimal paths taking into account the nominal and available bandwidth as well as current delays while delivering data packets or control information. These routers should have a well-defined relationship between: 1) data rate and the available throughput of virtual channel, 2) routing policy and current network topology, and 3) routing algorithms and characteristics of transport, network, and data link layer protocols.

Creating a mathematical model of such future ITS is complicated by the large number of interdependent variables. In our work, we have used simulation methods and Network

 ISBN: 978-1-61208-318-6

Simulator 3 (NS-3) [7] for verification of the proposed approaches. The NS-3 simulator version 3.16 does not have a premade solution for creation of a multiprotocol node that serves as a gateway between different wireless technologies. Building of simulation model for mobile communication network requires realization of a multiprotocol node model that functions as message router between MESH communication networks and stationary infrastructure [4]. Realization of the multiprotocol node was made on the base of a "spot-to-spot" virtual point that enables intermediate interaction between interfaces. The following modules of NS-3 were used to implement the model with the multiprotocol node:

- 802.11s interface model. Implementation allows us to use the Hybrid Wireless Mesh Protocol (HWMP) route protocol in proactive and reactive modes, in addition we use Optimized Link-State Routing (OLSR), Ad hoc On-Demand Distance Vector (AODV), Destination-Sequenced Distance Vector routing (DSDV) protocols for wireless MESH networks;
- Implementation of routing protocol models in wireless networks HWMP, OLSR, AODV, DSDV;
- FlowMonitor is a module of network traffic statistics collecting and processing;
- WireShark is an analyzer of computer network traffic;
- NS-3-highway-mobility is a model of vehicular traffic.

The simulation model allows to combine different routing protocols, network interfaces, and vehicles' traffic models. Simulation result is the set of xml-files generated by FlowMonitor module.

For experimental researches there was developed a specialized packet technology to initialize model's parameters and to realize the change of parameters during the experiment. These parameters are: vehicle's speed or trajectory change, routing protocol, transport layer protocol, connection throughput, number of nodes in the network, number of nodes transmitting data simultaneously, size of packets, packet loss rate in the communication channel [4]. While modeling for each node in the network the following characteristics are registered: packet's send and receive timestamp, packet loss rate, packet size, source and destination IP addresses. Output stores as the xml-files for future analysis. A simulation process with prescribed parameters allows researching the most dynamic periods of the MESH network existence (short time of the network static life, wide range of network traffic intensities, high intensity of route relocation).

Various sets of initialize parameters allow us to analyze different kinds of MESH and cloud-oriented states. For example, an estimation of security level for data transfer, actual speed of data transfer supported by the network, and average time of connection between the network mobile node and cloud-oriented environment.

In our researches, we have done two types of experiments:

1. Influence of the routing protocols on data transmission rate. We have considered OLSR [8], DSDV [9], AODV [10], and HWMP [11] routing protocols. As a source we've used UDP traffic with 8, 32, 64, 128, 512, 1024, 2048 Kb/s throughput. There was only one node with LTE interface.

The best results with high intensity flows are shown by AODV, DSDV protocols. To transfer short messengers with low intensity it is better to use HWMP.

2. Influence of vehicle traffic characteristics on the packet loss rate

2.1. HWMP, OLSR, AODV, and DSDV routing protocols were used. Network bandwidth was 8 – 2048 Kb/s. There was one node with LTE interface. Packet size was 1024 bytes. A number of vehicles was 8 or 16 for 800 meters of the road. Traffic speed was from 10 to 100 miles per hour (MPH). An actual packet loss rate was determined by Wireshark.

The reliability of message delivery was evaluated in a high dynamics of the network structure. Changes occurred at least once per second. Under these conditions, the intensity of the routing protocols was high and the packet loss rate was significant.

Fig. 4 shows the packet loss rate transmitted from the network 802.11s node to the cloud from the protocols used by the wireless routing and data transfer speed. Packet loss rate ranges from 7 to 46 percent. Packet losses increase with increasing transmission speed. By increasing the transmission speed twice, packet loss increases three times due to the broadcast routing requests. The greatest losses occur while using OLSR protocol.

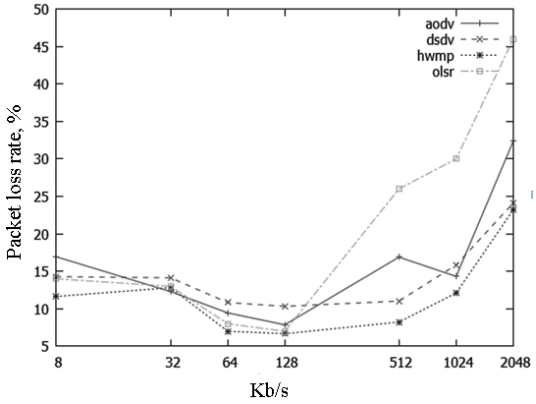


Figure 4. Packet loss rate from data transmission rate.

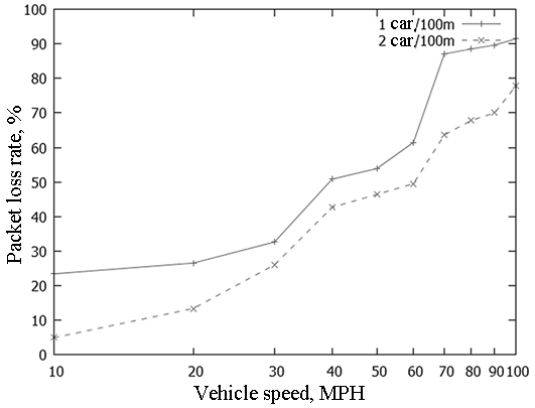


Figure 5. Packet loss rate from vehicle speed

2.2 Fig. 5 shows that for one or two vehicles that car moves at a speed of 100 MPH packet loss rate is 78-92%. The reason is that vehicles are in radio visibility zone for very short time, which is not enough to establish connection with the cloud environment. By increasing the number of cars on the road, we would decrease the packet loss rate.

 ISBN: 978-1-61208-318-6

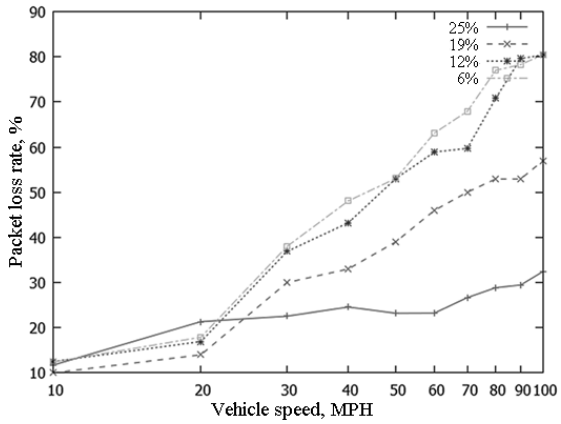


Figure 6. Packet loss rate while transmitting emergency message.

Fig. 6 shows the packet loss rate while transmitting emergency message from the vehicle on the strait road 800 meters long. Packet size is 1024 bytes. Number of cars is 16. Vehicle speed ranges from 10 to 100 MPH. The number of nodes with LTE interface is 1, 2, 3, and 4 (which corresponds to 6%, 12%, 19%, and 25% of all cars).

According to the received results (Fig. 6) packet loss rate is greater than 40%, while there are less than 25% of cars have LTE and speed is greater than 50 MPH. In this case, it is better to use alternative communication channels between vehicle and cloud environment.

IV. SECURITY SERVICES FOR CLOUD-ORIENTED ITS INFRASTRUCTURE

Now, we will discuss the organization of bilateral access control provided for the mobile and fixed components of the ITS. Due to the dynamic nature of cloud environment, we propose to realize the services' integration by using the expanding to cloud applications an existing version of the patented (US 7281129, RF 2214623) firewall decision. This approach allows dynamic changing access rules and operates in so called addressless or stealth modes using security policy semantic as an invariant for any interconnection between vehicles and ITS resources. An operational form of the proposed service can be represented as a set of traffic filters applied to each virtual connection.

The development of cloud computing environment requires new approaches to provide information security [12]. These requirements come from the need to consider the dynamic nature of the processes allocation of computing and network resources in the configuration of a virtual machines, which are the basic components of modern service infrastructure. In this section, we discuss an approach to the firewalls' configuration, by which are implemented control access policies to the cloud resources. Semantics of filtering rules and semantics of access policy must match or be close in a sense of chosen criteria to fulfill the requirements of information security for all possible configurations of cloud environment. Since the parameters of virtual machines that implement the application services are allocated dynamically the active filtering rules should also be changed during operation. In these circumstances, the traditional manual control settings of firewall rules in according to the access policy become impossible.

In such a dynamic environment as IaaS, the most stable part of the information relations is specifications of access policy. These specifications are special type of metadata that reflect the semantics of access strategy. Clearly, this strategy is not changing depending on the dynamic reconfiguration of available resources, so it can be regarded as a functional invariant of cloud services. Therefore, it is especially important to develop methods of automatic configuration of filtering rules and adaptation their parameters to the current state of cloud infrastructure, which can be viewed as carrier of the mentioned above invariant [6]. The formal model of cloud environment includes several parameters, namely:

$$\theta = \langle U, R, P, C \rangle, \quad (5)$$

where $U = \{u_i\}$, $i = 1 \ldots n$ is a set of cloud system's users. R is a set of roles $R = \{r_j\}$, $j = 1 \ldots m$. And each role is a set of privileges: $r_j = \{p_k\}$, $k = 1 \ldots l$, $r_j \subset P$, C – is a set of user sessions, which are presented by virtual connections between data source and destination in a cloud, P – set of privileges in the following form: $p_k = \{u, rul\}$, where $u \in U$ is a user of a cloud system, who is running information service (e.g., web application); $rul = \{r_g\}$, $g = 1 \ldots h$ – is a set of rules, which identify network application. The rule consist of the following parameters: $r = \langle transport, port, protocol, ext \rangle$, where *transport* is a transport layer protocol (e.g., tcp or udp), *port* – is a number of tcp or udp port, *protocol* – is an application layer protocol, *ext* – are additional parameters for application protocol.

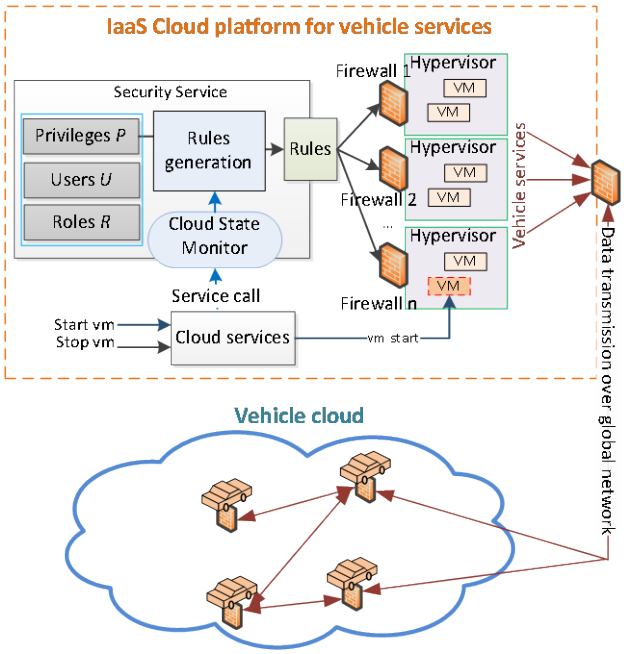


Figure 7. Security platform for cloud oriented ITS.

The example P is formed by the following parameters: *{user:*"Ivan", [{*transport:*"TCP", *port:*"80", *protocol:*"HTTP", *ext:*[{*method:*"GET"}]}]}. This privilege allows access to web servers, which are running on port number 80 using HTTP protocol and HTTP method GET to virtual machines that below to the user named Ivan. In order to continue our formalization, the state of cloud system is presented in the following form:

$$State = \{vm_i\}, i = 1..n, State \subset VM \times U, \quad (6)$$

 ISBN: 978-1-61208-318-6

where *U* is a set of cloud users, *VM* – is a set of IP addresses of running virtual machines in a cloud. *State* is a set of addresses of running virtual machines with labels of users who started virtual instance.

It is necessary to reconfigure security system each time when the *State* of a cloud system changes. Proposed security system consists of a group of firewalls and the security service, which is integrated with the cloud controller (see Fig. 7). This approach allows translating RBAC model of security policy to the set of filtering rules according to the state of cloud environment. As explained above when the *State* changes all firewalls on a mobile and fixed parts of ITS receive a message "new virtual machine with privileges *P* started" or "virtual machine with privileges *P* stopped" from cloud controller and then generate new filtering rules while keeping semantics of security policy.

To provide a consistent set of rules we consider operation $gen(a_s, a_o, p)$, which translates privilege delegated to virtual machine with ip address a_s into the object (vehicle service), which is running in virtual machine with ip address a_o:

$$(\mathrm{p} = \langle \mathrm{u_s}, \{\mathrm{rul_i}\}\rangle) \xrightarrow{\mathrm{gen}} \{\langle \mathrm{a_s}, \mathrm{a_o}, \mathrm{rul_i}\rangle\}, \mathrm{i} = 1..\mathrm{n}, \qquad (7)$$

All services are related with specific virtual machines and when a user requests new service ITS launches a new virtual machine and its security subsystem, inspects user's privileges, and generates filtering rules, which are running in another virtual machine. A stealth mode allows implementing the information protection system in a form of dedicated security domain. This domain can be quickly adapted to the current state of the network infrastructure and scaled if necessary to achieve seamless integration without reconfiguration of current ITS routing policy.

The proposed IaaS computing platform with integrated security is implemented in telematics department of the Saint-Petersburg State Polytechnical University and operates for ITS and other services. This platform is built using OpenStack services with custom proprietary software components. The platform installation is fully automated by Chef scripts and it is possible to install all services in a few hours on standard hardware. Our secure cloud computing test bench is available at the following address: http://cloudlet.stu.neva.ru.

V. CONCLUSION

The results of this research can be summarized as a perspective way to expand opportunity for practical implementation of future ITS within concepts "Internet of Things" and "Smarter Planet".

We have proposed a formalization of the routing problem that provides the opportunity to develop a constructive multi-functional hierarchical model of ITS, which could implement different classes of vehicle's services, including transfer of the special class of emergency messages.

As the part of our work, we have developed a simulation method allowing to combine the hierarchical network model with a specific structure of the urban transport network to select the optimal parameters of telematics services in the virtual network nodes for the delivery of emergency messages. The paper presents the results of the routing protocols' choice using simulation modeling in NS-3.

The proposed decisions can be used to reduce traffic congestion and emergency incidents, to support multi-protocol interactions, and to provide predictable real-time performance for different kinds of end-user applications, that needs to:

- Sharing data between mobile objects and fixed infrastructure nodes;
- Routing messages in multiprotocol mode;
- Delivering urgent and delay tolerance information;
- Integrating security services between ITS applications.

Information exchange technologies developed in this work are implemented in the international space experiment "Kontur-2" on board the ISS [13].

Our future research will be focused on applying a cloud computing paradigm to develop adequate to the times communication services and safer requirements for the future generation of ITS.

ACKNOWLEDGMENT

This paper funded by Russian Ministry of Education and Science and RFBR grant 13-07-12106. This research was supported by a grant from the Ford Motor Company.

REFERENCES

[1] http://www.ibm.com/smarterplanet/now [retrieved: Jan 2014].

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things", Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, New York, NY, USA, 2012. ACM, pp. 13-16.

[3] V. Zaborovski, M. Chuvatov, Ol. Gusikhin, A. Makkiya, and D. Hatton, "Heterogeneous Multiprotocol Vehicle Controls Systems in Cloud Computing Environment", In 10th International Conference on Informatics in Control, Automation and Robotics (ICINCO), SciTePress, 2013, pp. 555-561.

[4] M. Kurochkin,V. Glazunov, L. Kurochkin, and S. Popov, "Instrumental environment of multi-protocol cloud-oriented vehicular mesh network", In 10th International Conference on Informatics in Control, Automation and Robotics (ICINCO), SciTePress, 2013, pp. 568-574.

[5] S. Bitam and A. Mellouk, "Its-cloud: Cloud computing for intelligent transportation system", In Global Communications Conference (GLOBECOM), 2012 IEEE, pp. 2054-2059.

[6] V. Zaborovsky, O. Zayats, V. Mulukha, "Priority Queueing With Finite Buffer Size and Randomized Push-out Mechanism", Proceedings of The Ninth International Conference on Networks (ICN 2010), IEEE Computer Society, 2010, pp. 316-320.

[7] http://www.nsnam.org/ [retrieved: Jan 2014].

[8] http://www.ietf.org/rfc/rfc3626.txt [retrieved: Jan 2014].

[9] Perkins Charles E., Bhagwat Pravin: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers, London England UK, SIGCOMM 94-8/94.

[10] http://www.ietf.org/rfc/rfc3561.txt [retrieved: Jan 2014].

[11] S.M.S. Bari , F. Anwar, M.H. Masud, "Performance Study of Hybrid Wireless Mesh Protocol (HWMP) for IEEE 802.11s WLAN Mesh Networks", In International Conference on Computer and Communication Engineering (ICCCE), 2012, pp. 712-716.

[12] V. Zaborovsky, A. Lukashin, S. Kupreenko, and V. Mulukha, "Dynamic access control in cloud services", In Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on, pp. 1400-1404.

[13] V. Zaborovsky, A. Kondratiev, V. Muliukha, A. Silinenko, A. Ilyashenko, M. Filippov, "Remote Control Robotic Systems in "Kontur" Space Experiments", Informatics, Telecommunication and Control, Politechnical University, Saint-Petersburg, Russia 6(162), 2012, pp. 23-32 (in Russian).

 ISBN: 978-1-61208-318-6

A Technique to Mitigate the Broadcast Storm Problem in VANETs

Manoel Rui P. Paula, Daniel Sucupira Lima, Filipe Maciel Roberto,
André Ribeiro Cardoso, Joaquim Celestino Jr.
Computer Networks and Security Laboratory (LARCES)
State University of Ceará (UECE)
Fortaleza, Brazil
{manoel.rui, daniel.lima, filipe, andrec, celestino}@larces.uece.br

***Abstract*—Vehicular Ad Hoc Networks (VANETs) are networks formed by vehicles using the wireless medium as communication link for data transmission and reception. In this network, vehicles can transit on the roads with high speed and thus provide a high dynamism in its topology. This may cause the connectivity between vehicle lasts a short time. So, many applications in VANETs, which need to disseminate data traffic seek a fast and efficient diffusion mechanism. The forwarding broadcast mechanism is generally used to accomplish this task. However, a poorly elaborated mechanism for disseminating messages can flood the network with redundant data and increase the number of collisions due to disputes between vehicles for accessing the medium. These problems are usually known as broadcast storm problem. Thus, this paper proposes a probabilistic technique for mitigating the broadcast storm problem through a game from the Games Theory: The Volunteers Dilemma. In order to explore the equilibrium in the game and also evaluate the network technique performance some simulations using the Network Simulator 3 (NS3) were performed. The results showed that the technique presented a good delivery rate of packets and little loss of data with high vehicular densities. However, it was found that even without an overwhelming number of transmissions, a large amount of redundant information was noticed.**

Keywords-VANET; volunteers dilemma; routing protocol; broadcast storm;

I. INTRODUCTION

Vehicular networks known as VANETs are networks that show different characteristics from other wireless networks, such as, the Mobile Ad-hoc Networks (MANET) [1]. Both networks are wireless and self organized, in which their own nodes order and provide services. Although VANET is a special case of MANET, one of the main differences between them is their mobility patterns [2]. Since the key elements in a VANET are basically cars that communicate with each other through a wireless network, the direction and movement of the vehicles are usually limited by the dimensions of the way. Unlike VANETs, in MANETs the nodes can describe random trajectories. The main characteristics that distinguish VANETs from MANETs are found in [3].

In the environment of wireless vehicular networks, protocols that are designed to broadcast, transport, deliver or route messages from an application should be concerned with the special features that are in VANETs [3] [4]. Protocols originally designed for other types of networks generally have lower performance when applied in VANETs, since their characteristics and problems are different. Many applications in VANETs are designed to benefit all network elements, such as security applications in traffic to prevent collisions among vehicles [5]. Other applications in the same category which are supposed to send messages to all other nodes in the network use broadcast protocols. The use of a broadcast protocol in this case is a good strategy to disseminate data because the vehicles do not need to know an address and a route to a specific target [6]. However, the forwarding of messages following a broadcast protocol poorly prepared, with an excessive number of broadcasts, flood the network with duplicate messages and causes infinite loops of retransmissions. Since the network is wireless, vehicles share the same link and can compete for accessing the medium in a broadcast protocol for VANETs. This fact is especially true when a vehicle receives a specific packet and decides to relay it to other close vehicles. Recipient vehicles, which decide to relay the packet received in regular equal times cause huge amounts of collisions, consequently this makes the information transmitted by the packet maker vehicle is not passed on to further vehicles away from it. These problems are generally referred to as the broadcast storm problem [7] [8].

Thus, a smart mechanism that uses a technique to disseminate messages which are sent from the traffic generator vehicle and can be achieved by all neighbouring nodes is needed. In order to have the messages achieving vehicles out of the range of the transmitter vehicle, it is necessary that the intermediate vehicles forward these incoming messages. Since some protocols send messages only for management purposes, retransmitting these messages may cause a waste of processing, bandwidth, and a longer delay to access the medium controlled by the link layer protocols. Applications that require urgency, so that their messages reach the other nodes in the network would suffer a greater impact concerning a service break in the broadcast storm [7]. So, it is clear that the implementation of a broadcast mechanism poorly designed worsens the broadcast storm problem, overloading the network unnecessarily.

The other parts of this paper are organized as follows: Section 2 will show how routing protocols can be classified in VANETs. Section 3 focuses on main broadcast protocols that worked as a stimulus for formulating probabilistic broadcast protocol in this work. Section 4 shows the modeling of the Volunteer's Dilemma [21] game in order to produce a broadcast protocol for VANETs. In Section 5, the results

 ISBN: 978-1-61208-318-6

of experiments between two probabilistic protocols will be shown: the first comes from Quantal Response Equilibrium (QRE) [22] (proposed in this work) and the second comes from the Nash equilibrium [18]. Finally, Section 6 will discuss the conclusions and future work.

II. CLASSIFICATION OF ROUTING PROTOCOLS IN VANETS

Most routing protocols in the literature proposed for VANETs [6] [9] focus on more specific characteristics, making these protocols become quite limited. Thus, for a routing protocol it has a higher efficiency, some main features must be considered.

In [10], the essential features that one routing protocol must have are shown, some of them are the most important: the protocol must be dynamic, and acting reactively creating routes on demand. Another important feature is that the routing protocol must be scalable in such a way that the routing protocol must show good performance in scenarios with low and high quantity of vehicles. In order to achieve a better performance, a protocol should have mechanisms to know the network topology, even after the vehicles change their positions. Many times, it facilitates developing solutions to situations of broken connections. And lastly, a good feature for routing protocols is to provide a larger time of connectivity between vehicles. This latter feature is important for routing because it provides the required time to complete the calculation of routes taken by the protocol.

Through several researches on routing on wireless vehicular wireless networks, many protocols have been reported in scientific communities. However, it is difficult to find protocols that suit the different situations and scenarios. Many routing protocols in VANETs are designed in order to troubleshoot specific networks, thus these protocols may have similar properties and features. Not only by the protocols features, there are several ways to classify the routing protocols in VANETs. One can classify them by the techniques used, routing information, service quality, routing algorithms and others [9].

Some authors tend to classify routing protocols following a common genre. Works, such as [11], [12], and [13], classify protocols based on the use of techniques and particular characteristics in five classes: position, grouping, geocast routing, topology, and broadcast. However, other authors prefer to classify them in relation to their routing strategies, it means proactive or reactive [14]. In other scientific papers, the authors classify protocols regarding the information contained in the packet based on topology or geographic information [15].

Deepening the classification of protocols based on the strategies of transmissions, Lin [16] presents some of the main mechanisms for disseminating data in VANETs as well as main protocols in their category in the literature. In his work, he classifies routing protocols for disseminating information into three major groups: Unicast, Multicast, Geocast, and lastly, Broadcast.

Similar to Lin [16], Panichpapiboon and Pattara-Atikom [6] also classify the protocols regarding the main techniques for disseminating information on VANETs. However, the attention is directed to the broadcast protocols. In his work, the classification of broadcast protocols are done basically in two main categories: Multi-hop and Single-hop Broadcast.

Fig. 1 shows clearly the protocols broadcasts classification made by [6]. The figure also shows the main techniques used by the protocols. Some of these protocols are presented in the next section.

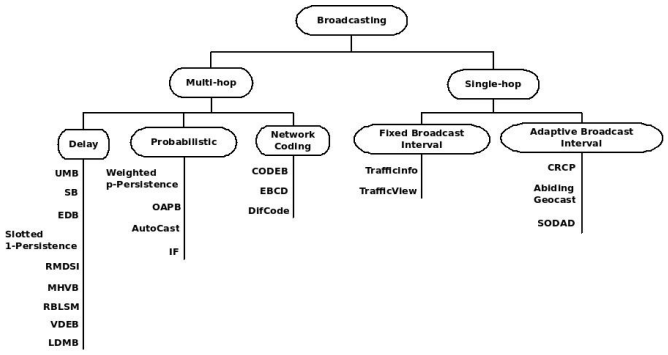


Figure. 1. Classification of broadcasts protocols in VANETs. Adapted from [6].

III. BROADCAST PROTOCOLS

Here, after some protocols presented in [6] will be shown, their main features will be discussed. The focus remains in some protocols which use techniques based on timers and probability, since parts of their approaches worked as inspiration for the development of this work.

The first protocol presented is the Weighted p-Persistence [7]. It is a probabilistic protocol. Protocols based on this category are characterized for deciding forwarding the received information by a probability p. In the weighted p-Persistence protocol, when a vehicle receives a packet via broadcast for the first time, it bases the need of rebroadcasting on the distance between the packet source node and it. The distance between them can be obtained through the receivers position and the transmitters position that is inserted into the sent packet. When a vehicle receives a broadcast packet it takes the decision to rebroadcast information with the probability (1).

$$p = \frac{D_{ij}}{R} \tag{1}$$

in which D_{ij} is the distance between the receiver i and the transmitter j, and R is the transmission range. The great advantage of using this approach is that it gives a certain priority to the vehicles near the radius edge R, what may provide greater achievement of packets transmitted on the network. However, this obtained probability does not consider the number of vehicles in the network. When the network shows a high density of vehicles, there may be a large number of vehicles with a high probability of forwarding and thus rise to a large number of retransmissions. The second protocol is called Irresponsible Forwarding (IF) [17], and it is also a probabilistic protocol. Similarly to the protocol Weighted

 ISBN: 978-1-61208-318-6

p-Persistence [7], the forwarding probability is built on the distance between the transmitter and the receiver. Although it also considers the density of vehicles. Thus, when a vehicle receives a broadcast packet, it decides to relay the information with the probability (2).

$$p = e^{-\frac{\rho_s (z-d)}{c}} \quad (2)$$

in which ρ_s is the vehicular density, z is the transmission range and d is the distance between the transmitter and the receiver vehicle. The parameter c is a coefficient with value greater or equal to one which is set to regulate the curve of the probability function. The main idea inserted in the formula is the fact that vehicles further from the transmitter have a higher probability of relaying information compared to vehicles which are closer to the transmitter.

The third protocol is the Slotted 1-Persistence Broadcast [7]. This protocol is based on timing. These protocols are characterized by retransmitting the information to other vehicles in the network after waiting a given time. It is expected that the waiting time set in the vehicles is different so that the vehicles far from transmitter are prioritized, it is, the closest vehicles to the edge in the transmission range will have a much shorter waiting time than other vehicles that are located closer to the transmitter.

So, the protocol Slotted 1-Persistence Broadcast uses a strategy using sectors to prioritize the vehicles waiting time concerning the distance. The transmission range of a vehicle is divided into sectors of times in which sectors more distant from the transmitter have a shorter waiting time than the ones closer to the transmitter. Thus, when a vehicle receives a broadcast packet, it computes its waiting time defined by the sector time following (3).

$$T_{S_{ij}} = S_{ij} * \tau \quad (3)$$

where τ is the propagation time of a jump between vehicles and the average waiting time for accessing the medium. And S_{ij} is the number of the sector defined by the distance between the transmitter and receiver. The sector number is calculated following (4).

$$S_{ij} = N_s \left(1 - \left\lceil \frac{min(D_{ij},\ R)}{R} \right\rceil \right) \quad (4)$$

where D_{ij} is the distance between the transmitter i and the receiver j, R is the transmission range, and N_s is the total number of sectors previously defined. The number of sectors must be wisely chosen since as the network becomes denser, there is the possibility that many nodes in the same sector have the same waiting time. This makes they transmit at the same time resulting in several collisions between packets.

Finally, a probabilistic protocol derived from Game Theory [18] [19] stands out over. This work worked as the main inspiration for the design of the protocol proposed in this work. In [20], a model using the symmetric game based on Volunteer's Dilemma is made [21] in order to mitigate the broadcast storm problem. In this work, we calculated the probability of a player volunteer derived from the Nash equilibrium [18] [19] with aiming to make the decision of transmitting a broadcast message by a vehicle in the network. The probabilistic protocol also incorporates some other strategies, such as the use of sectors and timers adopted by the Slotted 1-Persistence Broadcast protocol to prioritize the waiting time for transmission of a packet by a receiving vehicle over the distance from the source vehicle.

IV. VOLUNTEERS DILEMMA AS A BROADCAST PROTOCOL

Taking as a model the Volunteer's Dilemma [21] game using its symmetric version, it is possible to adapt it to the VANETs scenario in which a given vehicle in the network to which the network receives a broadcast message and decides to retransmit it to the others based on the probability of volunteering modeled by the game. Creating an algorithm for such purpose can be explored using the parameters involved in the game.

Let N be the number of vehicles participating in the game and all of them have a set of pure strategies, it means, each vehicle chooses to transmit or not transmit the received message. The decision of not broadcasting a message taking into account the fact that at least one vehicle has already made that decision, provides the greatest payoff B. The vehicle which decides to broadcast the message will pay a cost C and thus result in a payoff $B - C$. And in the case any vehicle does not decide to relay the message a minimum payoff of M will be shown. The representation of game in the regular way is shown in Table 1.

TABLE I
NORMAL FORM OF THE GAME.

	AT LEAST ONE FORWARD	ALL QUIET
FORWARD	$B - C$	$B - C$
QUIET	B	M

With the model presented by Goeree et al. [22], to insert noisy behavior and aversion to unequal gains for QRE equilibrium, it is possible to define a probability p of a vehicle to forward the incoming message to its neighbors following (5).

$$p = \frac{exp(\lambda \pi_v)}{exp(\lambda \pi_v) + exp(\lambda \pi_n)} \quad (5)$$

where $\pi_v = B - C - \alpha(1-p)C$ and $\pi_n = B(1-(1-p)^{N-1}) + M(1-p)^{N-1}$ with the precision parameter λ and aversion parameter α estimated through experiments. It is assumed that the cost C and the benefits B, M are properly defined.

As the game is the symmetric Volunteer's Dilemma, then the relationship C/B for vehicles belonging to the game should be similar. However, it is necessary to give a priority to the vehicles further away from the transmitter so that the message has a longer range with the vehicles in the network. Thus, a strategy using sectors similar to the one addressed by

 ISBN: 978-1-61208-318-6

Wisitpongphan et al. [7] and Roberto et al. [20] was adopted to prioritize the vehicle through space.

The transmission range R of the transmitter vehicle is divided in an amount N_{sect} denominated number of sectors. The numbering of the sector S happens in descending order tank into consideration the message transmitter, ie the sector most distant from the transmitter has value $S = 1$ and is labeled S_1, the second sector from the transmitter is labeled S_2 and so on until the nearest transmitter sector S_N. Each sector also limits some area to which vehicles may be passing. Thus, the length of the sector is defined as S_{len} and can be easily found using the transmission range R divided by the number of sectors N_{sect}. It can be formally defined as shown in (6).

$$S_{len} = R/N_{sect} \tag{6}$$

Since every vehicle in the network is transmitting and receiving ad messages (beacon messages) stating their mobility pattern, any vehicle that receives a broadcast message has the necessary information obtained to calculate the distance between it and the transmitter. That distance will be called the D_{tr}, which can be interpreted as the distance from a transmitter t from a receiver r. So, a vehicle can discover in which sector S_i it is compared to the transmitter performing the following computing (7).

$$S_i = \lfloor (N_{sect} + 1) \; - \; (D_{tr}/r) \rfloor \tag{7}$$

The modeling of costs and benefits is highly related to the transmission range of the vehicle and in which sector they are located. Thus, for a vehicle having a transmission range R equal to 1000 meters and the number of sectors N_{sect} equal to 5, vehicles located in the most distant sector from the transmitter will have benefits equal to 1000. Vehicles located in the second most distant sector from the transmitter will have their benefits equal to 800 because it is the result of a proportional reduction in the size of the length of the sector. The same idea is applied to the second most distant sector from the transmitter and so on until the closest section to the transmitter. So, a vehicle will know its benefit B_i through the sector it is in relation to the transmitter. That is, it will know its benefit calculating (8).

$$B_i = ((N_{sect} \; - \; S_i) \; + 1) \; S_{len} \tag{8}$$

To avoid the likelihood that at least one vehicle transmits the received message be reduced to zero due to cost and benefit have very similar values, resulting in a cost benefit close to 1, this cost is modeled as a function of the least benefit. That benefit is provided by the closest sector from the transmitter. The cost obtained is considered the same for all sectors. Then, the cost C can be set following (9).

$$C = B_{min} \; q \tag{9}$$

where B_{min} is the lowest benefit provided by the nearest sector from the transmitter and $0 < q < 1$ is a reduction factor of the lowest benefit. For example, if the lowest benefit achieved B_{min} is equals 200 and the factor q is equal to 0.5, then the modeled cost for all sectors will be equal to $C = 200 * 0.5 = 100$. In this work, the value of M is modeled with zero. Fig. 2 illustrates a scenario in VANETs in which nodes in the network use the sector approach to model the costs and benefits of the Volunteers Dilemma game. The algorithm resulting from the proposed modeling Volunteer's Dilemma game is defined in Algorithm 1.

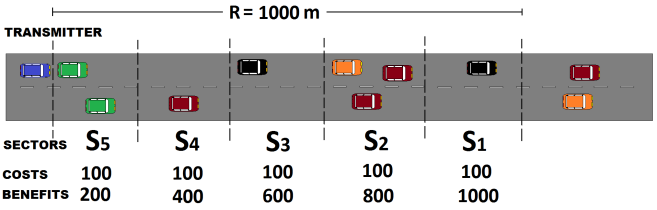


Figure. 2. Modeling of costs and benefits.

In: Packet received by broadcast
if *(The packet was received earlier)* **then**
  Discard packet;
else
  $S_{len} = R/N_{sect}$;
  $S_i = \lfloor (N_{sect} + 1) \; - \; (D_{tr}/r) \rfloor$;
  $B_i = ((N_{sect} \; - \; S_i) \; + 1) \; S_{len}$;
  $C = B_{min} \; q$;
  $\pi_v = B \; - \; C \; - \; \alpha(1-p)C$;
  $\pi_n = B(1-(1-p)^{N-1}) \; + \; M(1-p)^{N-1}$;
  $p = \frac{exp(\lambda \pi_v)}{exp(\lambda \pi_v)+exp(\lambda \pi_n)}$;
  n_{rand} = RAND(0, 1) ;
  if $(n_{rand} \leq p)$ **then**
    Forward packet;
Out: Transmission, or not, of packet received..

Algorithm 1: Proposed algorithm for broadcast forward.

In order to perform the calculations required for the probabilistic decision of retransmitting a packet received via broadcast, one vehicle must know some information of its neighbors. However, this decision is not made cooperatively but individually. Thus, a message exchange mechanism between neighbors becomes essential as a tool for the capture and dissemination of information in the neighborhood. For this reason, a management mobility mechanism was created as a separate module inspired by [23] and is also used by [20] in their work.

So, the game starts in each sector whenever a vehicle performs the spread of a packet via broadcast and vehicles located within sectors limited by the transmission range must make the decision to retransmit information for the other vehicles in the network or not. The game ends when all vehicles have taken their decisions.

Thus, the algorithm is applied when a vehicle in a sector receives a packet as input. If a packet is identified as received

 ISBN: 978-1-61208-318-6

then it is discarded. Otherwise, if it was received for the first time, then the receiver vehicle uses the information as the basis of its neighborhood table to calculate the values of the expected payoff of volunteer π_v and of not volunteering π_n to be applied in the formula of probability p. The number of players in a sector is obtained using the mobility management module. Considering the parameters λ and α estimated [22], involving the game parameters, the vehicle transmits the packet with probability p.

V. RESULTS

The experiment of this work is a freeway scenario with two lanes where vehicles move with constant speeds in the same direction. (topology similar to Fig. 3). The transmitter vehicle is sending packets every second in the simulation time. When the receiver receives a message from the source message vehicle or coming from a relay, the implemented algorithm runs and decision to forward the message is taken.

Each vehicle has a transmission range of approximately one kilometer following the Nakagami propagation model, the radius is divided into five sectors and therefore the size of each sector is 200 meters. Thus, in order to evaluate the performance of probabilistic techniques in more realistic scenarios in which network connectivity depends on the distance between the vehicles, a modeling was performed where the inter vehicle spacing was exponentially distributed. All experiments were done using the NS3 [24] to evaluate the game performance in the network. The simulations were performed with the following vehicular densities: 10 (vehicles/km), 15 (vehicles/km), 30 (vehicles/km), 45 (vehicles/km) and 60 (vehicles/km).

The simulation parameter values were chosen according to levels of vehicular traffic (light, moderate and heavy) [7] and also induces behavior as the probability of a vehicle forward data, since the amount of vehicles in a sector and the cost-benefit determine these probabilities [21] [22]. The parameters were heavily influenced by the works of [7] and [20]. All simulation results were obtained with confidence level equal to 0.95 for the confidence interval of each of the averages of data obtained through measurements presented below.

For the result analysis, the probabilistic technique based on Nash equilibrium of the symmetric game based on the volunteers dilemma simply Nash Equilibrium [21], was call. This technique is similar to [20], as shown in Section 3. In addition, the technique based on the probabilistic balance QRE, also from the same game, called the QRE equilibrium [22].

Another observation to be made concerns the forwarding probability for each routing technique. Fig. 3 shows the behavior of the probability of retransmission in each sector to the Nash equilibrium and each density discussed previously. Eg vehicular density equals to 10 vehicles/km corresponds to the same vehicular density of 2 vehicles/sector since the radius is partitioned into 5 sectors and each sector has 200 meters. Similarly, Fig. 4 shows the behavior of the probability of retransmission in each sector for QRE equilibrium.

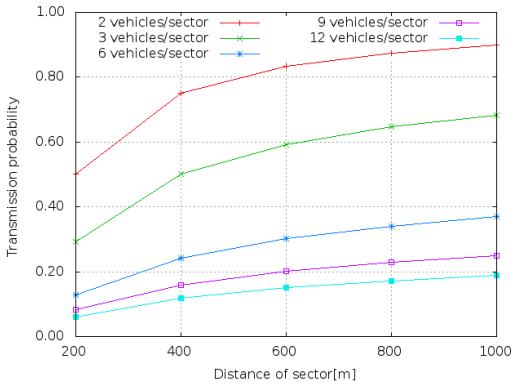


Figure. 3. Probability of retransmission of packets in each sector to Nash equilibrium.

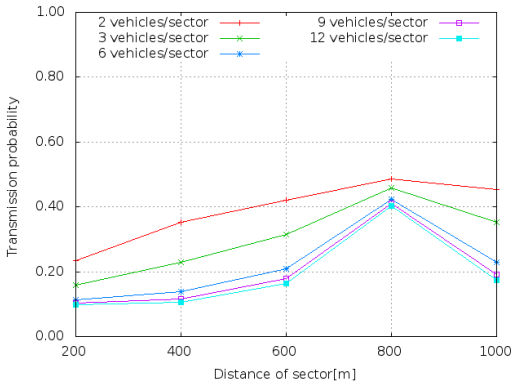


Figure. 4. Probability of retransmission of packets in each sector to QRE equilibrium.

Standard packet delivery rate: The expected analysis of this metric concerning both techniques is that the delivery rate increases as the vehicular density also increases. That is exactly what happens in the graphs shown in Fig. 5. In high vehicles densities, the ones which are located closer to the transmitter have a great chance to receive the data transmitted.

The graphs also show that the Nash equilibrium has a packet delivery rate greater than QRE equilibrium for lighter and moderate vehicular traffic. This is explained by the fact that the probability of routing in the Nash equilibrium is relatively high for small groups, thus more retransmissions are performed. However, when the number of players increases, the probability of volunteer decreases considerably. In the QRE equilibrium, the probability behavior of volunteering does not decrease a lot compared to the Nash equilibrium. Thus, for large groups, the probability of forward in the QRE equilibrium becomes higher than in the Nash equilibrium and therefore a larger amount of packets will be sent and received successfully.

Standard packet loss rate: In Fig. 6, the graphs show that the QRE equilibrium performance was worse than the Nash equilibrium in the first three densities. This happens because the probability of forwarding in the Nash Equilibrium is greater than the equilibrium QRE. It can also be noted that

 ISBN: 978-1-61208-318-6

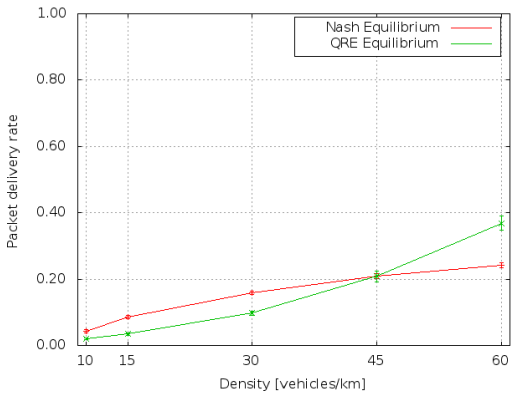


Figure. 5. Standard rate packet delivery.

the further away from the transmitter, the signal information becomes weaker, getting to a point in which the network device will no longer be able to receive the message. In larger networks, there is a greater tendency for this to happen and thus resulting in a higher packet loss. Another factor that influences the rate of packet loss is the amount of collisions during the broadcast storm.

With vehicular density equal to 45 vehicles/km, both techniques presented performances quite similar. But in a heavy vehicles traffic it can be noticed a reversal in performance techniques. As previously explained and reassured in the results from the rate of delivered packets, the probability of routing in QRE equilibrium remains higher than the Nash Equilibrium groups with lots of players. Thus, the packet transmitted by the source message vehicle has a great chance to reach the destination and few packets are lost in the network.

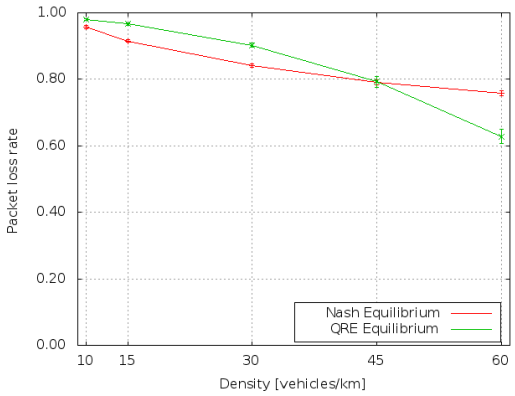


Figure. 6. Standard packet loss rate.

Number of duplicate packets: The greater the number of message retransmissions in the network it is expected a greater number of duplicate messages. The graphs in Fig. 7 shows the results obtained from both techniques to prove this fact. As we can see, the Nash equilibrium got better results than the QRE equilibrium in the following densities: 10 vehicles/km, 15 vehicles/km and 30 vehicles/km.

With a density of 45 vehicles/km there was a slight difference in the techniques performance where it was from there that the QRE equilibrium proved to be the worst. It may be noted that for the higher density, the number of duplicated packets in QRE equilibrium was slightly more than twice than at the Nash Equilibrium. A greater difference in the values shown from that point is justified because the vehicles are closer to each other. As the transmission range vehicle can cover a large number of neighboring vehicles any relay performed by any of these vehicles in high density networks will result in large amount of duplicate information.

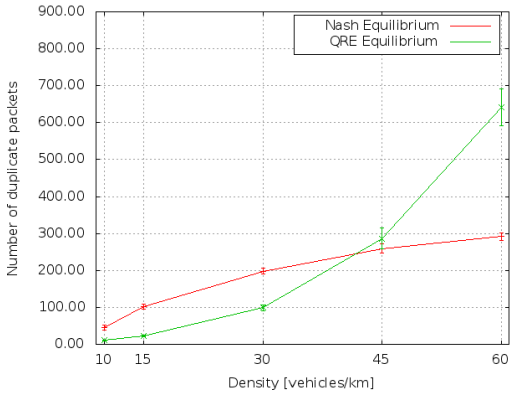


Figure. 7. Number of duplicate packets.

Link load: Finally, we have the average results from the load link shown in Fig. 8. These results strengthen what other metrics showed. The greater the amount of information transmitted by a given vehicle, the greater the amount of information received by receiver vehicles. The routing probability in both techniques also shows the same pattern.

In the lower densities of 10 vehicles/km, 15 vehicles/km and 30 vehicles/km the values were very small because losses may have happened. But this can be explained because the receiving vehicles are usually away from the transmitter vehicle and thus, for vehicles outside the range of the transmitter, the arrival of packets generated by the source depends on retransmissions carried out by vehicles within the transmission range.

Similarly to what happened with the other previous metrics, with density equal to 45 vehicles/km the performance of both techniques were very similar, mainly motivated by the behavior of their quite similar probabilities. And for the density of 60 vehicles/km QRE equilibrium keeps the routing probability greater than the Nash equilibrium and thus it gave a higher amount of transmissions and thus receptions. The analysis of the number of duplicated packets proves what was said and evaluated the load link results.

VI. CONCLUSION

In this work, a modeling of the game Volunteer's Dilemma was carried out, particularly the symmetric version, and this idea was based on Game Theory in a totally different environment, such as the vehicular networks. With the main objective

 ISBN: 978-1-61208-318-6

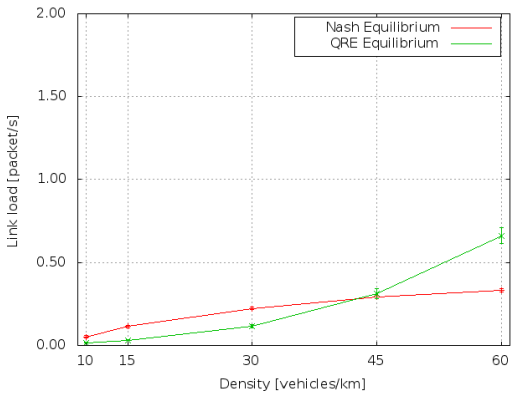


Figure. 8. Link load.

to mitigate the effect of broadcast storm in VANETs, a proposed probabilistic technique based on the QRE equilibrium was made.

From the results we can see the behavior and the characteristics involved in the game of volunteers dilemma. Particularly, we observed the effect of probability to volunteer in a Nash equilibrium stemmed from mixed strategies and the probability of volunteering in QRE equilibrium with additional parameter of aversion. As expected, the results strengthened the concepts in both equilibrium involved, since the parameters to setting the routing probability QRE equilibrium contribute to this probability decrease more slightly than in the Nash equilibrium.

The probability impact of a player volunteer using the probability stemmed from the QRE equilibrium provides some positive contributions. For example, when vehicular density is high, a delivery and loss rate becomes more satisfactory than in low densities. But a side effect in the same situation in a network with high vehicular density is that it can generate a higher rate of redundant information in the network.

Thus, the proposal in this work proved to be a very interesting and satisfying technique to spread information via broadcast in VANETs in order to mitigate the broadcast storm problem. In order to be a technique to be applied in a more robust dissemination protocol, some improvements can be made. A further study to reduce the packet loss rate caused by collisions can be made. In this sense, the use of timers can be inserted to prevent simultaneous transmissions. Another improvement that can be exploited is through modeling costs and benefits, verifying the ideal values in their modeling for each scenario. It is also possible to run some simulations with costs and benefits asymmetrically in order to prioritize vehicles further from the transmitter. Another way to prioritize vehicles can be made by exploiting the quality and strength of the signal coverage area of the transmitter vehicle. Thus, with this change, a greater amount of scenarios could be tested and compared with some protocols in literature in order to assess its efficiency and viability.

REFERENCES

[1] R. Kaur and M. K. Rai, “A novel review on routing protocols in manets,” Undergraduate Academic Research Journal (UARJ), 2012, pp. 2278–1129.

[2] R. Sharma, M. Halder, and K. Gupta, “Mobile ad hoc networks-a holistic overview,” International Journal of Computer Applications, vol. 52, no. 21, August 2012, pp. 31–36.

[3] B. Paul, M. Ibrahim, and M. A. N. Bikas, “Vanet routing protocols: Pros and cons,” International Journal of Computer Applications, vol. 20, no. 3, April 2011, pp.28–34.

[4] S. Kumari, “Survey on routing protocols in vanet,” International Journal of Wired and Wireless Communications, vol. 2, no. 1, 2013.

[5] V. R. Krishnan and T. Rajesh, “Vanet based proficient collision detection and avoidance strategy for cars using double-c curve movement algorithm,” IJCAIT, vol. 1, no. 2, 2012, pp. 81–84.

[6] S. Panichpapiboon and W. Pattara-Atikom, “A review of information dissemination protocols for vehicular ad hoc networks,” Communications Surveys Tutorials, IEEE, vol. 14, no. 3, 2012, pp. 784–798.

[7] N. Wisitpongphan, O. Tonguz, J. Parikh, P. Mudalige, F. Bai, and V. Sadekar, “Broadcast storm mitigation techniques in vehicular ad hoc networks,” Wireless Communications, IEEE, vol. 14, no. 6, 2007, pp. 84–94.

[8] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” Wireless networks, vol. 8, no. 2-3, 2002, pp. 153–167.

[9] M. Altayeb and I. Mahgoub, “A survey of vehicular ad hoc networks routing protocols,” International Journal of Innovation and Applied Studies, vol. 3, no. 3, July 2013, pp. 829–846.

[10] M. Barros, R. C. M. Gomes, and A. da Costa, “Routing architecture for vehicular ad-hoc networks,” Latin America Transactions, IEEE (Revista IEEE America Latina), vol. 10, no. 1, pp. 1411–1419, 2012.

[11] P. Bijan and M. J. Islam., “Survey over vanet routing protocols for vehicle to vehicle communication,” IOSR Journal of Computer Engineering (IOSRJCE), vol. 7, no. 5, 2012, pp. 1–9.

[12] S. Allal and S. Boudjit, “Geocast routing protocols for vanets: Survey and geometry-driven scheme proposal,” Journal of Internet Services and Information Security (JISIS), vol. 3, no. 1/2, 2013, pp. 20–36.

[13] R. Kumar and M. Dave, “A comparative study of various routing protocols in vanet,” IJCSI, vol. 8, no 1, July 2011, pp. 643–648.

[14] M. Vijayalaskhmi, A. Patel, and L. Kulkarni, “Qos parameter analysis on aodv and dsdv protocols in a wireless network,” International Journal of Communication Network & Security, vol. 1, no. 1, 2011.

[15] U. L. Kevin C. Lee and M. Gerla, “Survey of routing protocols in vehicular ad hoc networks,” IGI Global, no. 22, pp. 149–170, 2010.

[16] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, “Routing protocols in vehicular ad hoc networks: A survey and future perspectives,” J. Inf. Sci. Eng., 2010, pp. 913–932.

[17] S. Panichpapiboon and G. Ferrari, “Irresponsible forwarding,” ITS Telecommunications, pp. 311–316, 2008.

[18] J. Watson, Strategy: An Introduction to Game Theory. W. W. Norton & Company; 2 edition, 2007.

[19] R. B. Myerson, Game theory: analysis of conflict. Harvard university press, 2013.

[20] F. Roberto, J. Celestino, and H. Schulzrinne, “Using a symmetric game based in volunteer’s dilemma to improve vanets multihop broadcast communication,” Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on, 2011, pp. 777–782.

[21] J. Weesie, “Asymmetry and timing in the volunteerś dilemma,” Journal of Conflict Resolution, vol. 37, no. 3, 1993, pp. 569–590.

[22] J. K. Goeree, C. A. Holt, and A. K. Moore, “An experimental examination of the volunteers dilemma,” ESA Meeings, pp. 1–18, October 2005.

[23] J. Härri, C. Bonnet, and F. Filali, “Kinetic mobility management applied to vehicular ad hoc network protocols,” Comput. Commun., vol. 31, no. 12, 2008, pp. 2907–2924.

[24] (January, 2014) The network simulator. [Online]. Available: http://www.nsnam.org/

 ISBN: 978-1-61208-318-6

An Alleviating Traffic Congestion Scheme Based on VANET with a Function to Dynamical Change Size of Area for Traffic Information in Urban Transportations

Shinji Inoue, Yousuke Taoda, Yoshiaki Kakuda
Graduate School of Information Sciences
Hiroshima City University
Hiroshima, Japan
{shnj-496@, taoda@nsw.info., kakuda@}hiroshima-cu.ac.jp

Abstract— **The traffic congestion frequently occurs in urban transportations. Network services for alleviating the traffic congestion based on vehicle networks have been studied currently. Vehicle Information and Communication System (VICS) is one of the network services. It has been developed in Japan. However, since each vehicle can obtain global information on traffic congestion using VICS, all vehicles in the congested areas tend to move to non-congested areas. As a result, the non-congested areas become congested areas. To avoid such an oscillation between the congested and non-congested areas, we have proposed an Alleviating Traffic Congestion method (ATC). In the ATC method, each vehicle gathers traffic information in a limited area where the vehicle exists by using a Vehicle Ad hoc Network (VANET). The size of limited area is constant in the original ATC method. However, we consider that the size of the limited area should be adaptive depending on traffic condition. This paper proposes a modified ATC method, which can change the size of the limited area depending traffic condition. Through simulation experiments, this paper shows that the proposed method provides faster velocity and shorter trip time than VICS in the environments that traffic varies temporally and spatially, which occur in urban transportations.**

Keywords-alleviating traffic congestion; vehicle density; average velocity; VANET; simulation

I. INTRODUCTION

In order to alleviate the traffic congestion, the Vehicle Information and Communication System (VICS) [1] is famous in Japan. In VICS, ultrasonic vehicle detectors or other vehicle detectors, which are deployed by traffic control centers, report traffic information at the point where each detector is deployed to VICS center. The VICS center analyzes global traffic information reported by the vehicle detectors and generates traffic congestion information. The traffic congestion information is broadcasted by character multiplex, radio beacons, or optical beacons to vehicles. Vehicles which receive traffic congestion information can select driving routes where the routes do not go through congested areas. However, VICS depends on road-side infrastructure, for example vehicle detectors. If traffic congestion happens in an area where vehicle detectors are not deployed, VICS cannot broadcast the above traffic congestion information. Moreover, a research report [2] says that new traffic congestion would be caused when ratio of the number of vehicles in which a VICS terminal is installed to the number of all vehicles exceeds 50%. It is because vehicles in which a VICS terminal is installed would perform a similar behavior and the vehicles would pass through the same roadway segment.

Vehicle route management systems based on centralized servers have been proposed in [3] [4]. In [5], a distributed traffic navigation system has been proposed but road-side infrastructure should be required for each intersection.

Instead of global information obtained by VICS, this paper utilizes local information obtained by the Vehicle Ad hoc Network (shortly, VANET). There are lots of research results on collection of traffic congestion information using VANET. Shibata et al. [6] proposes a method to divide a transportation network into non-overlapped areas and to detect areas where congestion occurs in the transportation network using VANET. In the proposed method, each vehicle records not only the traffic congestion degree of roads through which the vehicle has passed but also the traffic congestion degree of roads through which the other vehicles have passed by exchanging it with them. As time proceeds, the entire traffic congestion information in the transportation network can be obtained by integrating the recorded information. Shinkawa et al. [7] proposes another method to obtain the entire traffic congestion information in the transportation network using cyclic vehicles such as fixed route buses. Yamashita et al. [8] proposes a method to suggest an appropriate route using the road-to-vehicle communication as follows. First, each vehicle sends route information to reach the destination to a central server located in the roads using the road-to-vehicle communication. Next, the central server, which received the route information from many vehicles, calculates an appropriate route to the destination and sends back the calculated route information using the road-to-vehicle communication. However, to our knowledge, there are no proposals which are based on the traffic congestion information collected by VANET, each vehicle independently decides a route to alleviate traffic congestion in the transportation network.

In [11], we have proposed an Alleviating Traffic Congestion method (shortly, original ATC method.) The original ATC method works on each vehicle. Also the method suggests driving route information to each vehicle based on traffic information in a limited area around the vehicle.

 ISBN: 978-1-61208-318-6

In the original ATC method, the size of a limited area is constant. However, we consider that the size of a limited area should be changed depending traffic condition. In this paper, we propose a modified Alleviating Traffic Congestion method (shortly, a modified ATC method.)

The rest of this paper is organized as follows. In Section II, we explain transportation engineering. Next, we propose the modified ATC method in Section III. Section IV shows results of simulation experiments. Finally, we conclude our paper in Section V.

II. TRANSPORTAION ENGINEERING

In transportation engineering, a stream of vehicles on a roadway segment is mainly characterized by three kinds of variables. They are traffic volume, density, and velocity. A roadway segment is a part of roadway where there exist neither junctions nor road forks in the part. Traffic volume is the number of vehicles which pass through a fixed measuring point per unit time. Density is defined as the number of vehicles per unit length of roadway. A velocity of each vehicle is individual. So, in order to characterize a stream of vehicles, time average velocity and space average velocity are calculated. Time average velocity is the average of vehicles velocity when each vehicle passes through a fixed measuring point during a fixed period. Space average velocity is the average velocity of the vehicles in a whole roadway segment at a fixed time instance. Both time average velocity and space average velocity would be influenced by traffic signals.

There exist some relations among traffic volume, density, and velocity. First, we mention that a relation between density and velocity. The higher density of vehicles is, the slower velocities of vehicles are. Relations between density and velocity are observed by many researchers and several relation models are proposed. In this paper, we adopt an equation of Greenshields model [9],

$$v = v_f(1-k/k_j) \quad (1)$$

where v is velocity, v_f is a parameter, which is called as free velocity, where a driver chooses velocity under a situation where no traffic flows affect the driver, k is density, and k_j is a parameter which is called as jam density which is the upper bound of density. Next, we mention that a relation between density and traffic volume. If density of vehicles is zero, traffic volume is obviously zero. While density of vehicles is less than some threshold, increasing the density causes increasing traffic volume. However, increasing the density makes velocities of vehicles slower. Consequently, when the density of vehicles is over the threshold, increasing the density causes decreasing traffic volume. At last, it makes velocities of vehicles to be zero because density of vehicles becomes saturated. Then traffic volume becomes zero. The threshold is called as critical density.

III. MODIFIED ATC METHOD

In this section, we describe the modified ATC methods.

A. Assumptions

In this paper, we assume the following matters on each vehicle.

- Each vehicle has a wireless communication function in order that such vehicles can form a VANET.
- Each vehicle has a GPS or a function showing an ID of roadway segment in which the vehicle currently exists.
- Each vehicle has road map information in order to show a receiving side vehicle's position to a road map.

Also, we assume the following matter on each roadway segment.

- Each roadway segment is not a one-way street, that is, each roadway segment is bi-directional way.

B. Outline of the Original ATC Method

In the original ATC method [11], since traffic congestion varies as time advances, each vehicle independently performs the following actions.

- Action 1: Each vehicle periodically broadcasts a request message through a VANET to obtain information of vehicles in the limited area around the vehicle.

For convenience, we call a vehicle which broadcasts the message a sending side vehicle. Also we call a vehicle which receives the message a receiving side vehicle.

All request messages have a same Time-To-Live (shortly, TTL) value. If TTL value of the request messages is equal to N, the request messages are re-broadcasted N times by receiving side vehicles.

- Action 2: Each receiving side vehicle replies a response which includes information about the vehicle (vehicle ID, velocity of the vehicle, roadway segment ID in which the vehicle exists, etc.)
- Action 3: The sending side vehicle receives responses from the receiving side vehicles and evaluates each roadway segments based on the responses.

To evaluate a roadway segment is estimating trip time where a vehicle goes through the roadway segment. In Action 3, each vehicle receives responses. Therefore, each vehicle knows density of vehicles in roadway segments. Following relations between density and velocity are well known. The higher density of vehicles is, the slower velocities of vehicles are. That is, each vehicle estimates velocity of vehicles in a roadway segment by density of vehicles in the roadway segment.

- Action 4: The sending side vehicle calculates a route for a destination of the sending side vehicle.

By the estimating trip time of roadway segments, each vehicle finds time-shortest path to a destination of the vehicle. A situation of each roadway (congested, non-congested) is varying as time goes by. In order to adjust current roadways' situations in real time, the above instructions are performed repeatedly.

 ISBN: 978-1-61208-318-6

C. The Modified ATC Method

Basic ideas of the original ATC method are as follows. First, the original ATC method leads vehicles to go through in time-shortest path. Next there might exist many roadway segment whose density of vehicles are small in the time-shortest path. As a result, vehicles which are in congested area would go to non-congested area.

However, suppose that the TTL value of request messages is small. In heavy vehicle traffic, the original ATC method could not find roadway segments whose density of vehicles is small. Also, it becomes fail to lead vehicles from congested areas to non-congested areas.

Conversely, suppose that the TTL value of request messages is large. Even in heavy traffic, the original ATC method might find roadway segments whose density of vehicles is small. However, the larger the TTL value is, the more vehicles know the low density roadway segments. Therefore, a large number of vehicles tend to gather in the low density roadway segments. As a result, a new traffic congestion would happen in the low density roadway segments.

We consider that the TTL value of request packets is adaptive to traffic conditions. A basic policy of the modified ATC method is to control the TTL value where the number of non-congested roadway segments which each vehicle can find is some fixed range (for example, the number of non-congested roadway segments is more than beta and less than gamma.)

Here, we describe the modified ATC method. The modified ATC method is same as the original ATC method except the followings.

After Action 3 of the original ATC method, the modified ATC method performs an adjusting TTL value operation.

Adjusting TTL value operation:

(1) Select responses which are sent from receiving vehicles in a half circle facing to a destination as depicted in Fig. 1. There exists Car X in the center of a circle. A destination of Car X is the corner of top right. In this case, the adjusting TTL value operation selects responses form a half circle of A.

(2) Count the number of non-congested roadway segments. (A density of vehicles parameter alpha is given. When density of vehicles of a roadway segment is less than alpha, the modified ATC method recognizes that the roadway segment is non-congested.)

(3) Suppose that the number is equal to K. If K is less than beta, increment TTL value by one. If K is greater than gamma, decrement TTL value by one. (Parameters beta and gamma are given. Beta is less than gamma. In order to avoid that TTL value becomes too large or too small, an upper bound and an lower bound of TTL value are set.)

IV. EVALUATIONS

In order to evaluate influence of parameters beta and gamma in the modified ATC method, we performed simulation experiments. For evaluating influence parameters, we performed the following combinations of beta and gamma, (15, 25), (20, 30), (40, 50), (25, 40), (10, 30), (30, 50), (20, 50). (The first element means beta. The second element means gamma.)

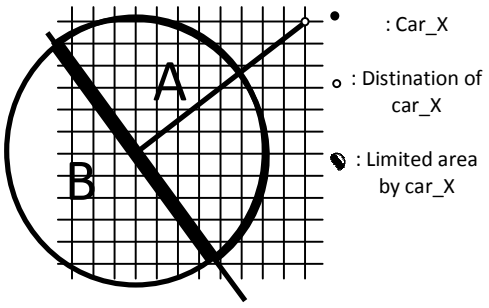


Figure 1. Area for selecting responses

In this simulation experiments, we adopt that congestion threshold density alpha is equal to 60 vehicles per a kilometer.

We performed one simulation experiments for each combination.

A. Simulation Environments

There exist no traffic flow simulators supporting inter-vehicle communications to our knowledge. So, we have developed a traffic flow simulator supporting inter-vehicle communications.

(1) Inter-Vehicle Communications

In the simulator, any two vehicles can communicate at any time in a single-hop communication if a distance between the two vehicles is less than a wireless communication range which is a given parameter. Moreover, the simulator does not take into consideration the communication delay time. In our simulation experiments, we set a wireless communication range 250 m.

(2) Setting Roadway Parameters

We have performed simulation experiments under the following road parameters. (See Table I)

(3) Setting Vehicle Parameters

(3-a) Control of Acceleration and Deceleration

The simulator prepares a function where each vehicle accelerates or decelerates based on a traffic flow model which is called as "Optimal velocity model" [10]. In this model, an optimal velocity is determined by a non-linear monotonic function of headway distance.

TABLE I. PARAMETERS ON ROADWAY SEGMENTS

Road network pattern	Square grid
Size	12 grid x 12 grid
Size of each grid	500 m x 500 m
Traffic signals	One for every intersection
Time period for red signal	67 s
Time period for yellow signal	3 s
Time period for green signal	60 s

 ISBN: 978-1-61208-318-6

TABLE II. TOTAL NUMBER OF THAT EACH ROADWAY SEGMENT BECOMES CONGESTED

beta, gamma	Total number of congestions
15, 25	24984
20, 30	22608
40, 50	19979
25, 40	20083
10, 30	19132
30, 50	18895
20, 50	16034

(3-b) Scenario of New Vehicles' Appearances

In these simulations, we prepared the same scenario where congested areas and non-congested areas appear and these areas relocate as time progresses. We omit the details of the scenario because of a page limit.

B. Results

Table II shows the results on total number of that each roadway segment becomes congested.

C. Considerations

The simulation results show that the number of congestion becomes small when the difference of beta and gamma is large. In the simulation of combination (15, 25), we consider that low value of gamma forces that TTL value becomes too small. In this case, changing operation of TTL value is influenced from value of gamma rather than traffic conditions.

Conversely, in the simulation of combination (40, 50), we consider that high value of beta forces that TTL value becomes large.

V. CONCLUSION AND FUTURE WORK

This paper has proposed the modified ATC method. In the modified ATC method, size of area for gathering traffic information is changeable depending on traffic conditions. For changing size of area, the modified ATC method introduces two parameters. Simulation results show that various combinations of parameters make a different influence for alleviating traffic congestions. In future work, we would like to analyze influences of parameter combinations for alleviating traffic congestion.

ACKNOWLEDGMENT

This research is supported by JSPS KAKENHI (Grant-in-Aid for Scientific Research (B), No.24300028) and MIC Strategic Information and Communications R&D Promotion Programme (ICT Innovation Creation R&D, No. 131408006).

REFERENCES

[1] VICS home page, http://www.vics.or.jp/ (in Japanese) [retrieved: December, 2013]

[2] I. Tanahashi, H. Kitaoka, M. Baba, H. Mori, S. Terada, and E. Teramoto, "NETSTREAM, a traffic simulator for large-scale road networks," R&D Review of Toyota CRDL, vol. 37, no. 2, 2002, pp. 47-53.

[3] T. Yokota, T. Nagai, K. Takahashi, Y. Kobayashi, N. Hamada, and M. Imaizumi, "Traffic flow optimization over a wide area," Proc. of Vehicle Navigation and Information Systems Conference, September 1994, pp. 193-197.

[4] K. Collins and G. Muntean, "A vehicle route management solution enabled by wireless vehiclar networks," Proc. of Infocom, April 2008, pp. 1-6.

[5] R. K. Guha and W. Chen, "A distributed traffic navigation system using vehicular communication," Proc. of 1st. IEEE Vehicular Network Conference, Octorber 2009, pp. 1-8.

[6] N. Shibata, T. Terauchi, T. Kitani, K. Yasumoto, M. Ito, and T. Higsashino, "A method for sharing traffic jam information using inter-vehicle communication," Proc. 3rd Annual International Conference on Mobile and Ubiquitous Systems Workshop, July 2006, pp. 1-7.

[7] T. Shinkawa, T. Terauchi, T. Kitani, N. Shibata, K. Yasumoto, M. Ito, and T. Higashino, "A technique for information sharing using Inter-vehicle communication with message ferrying," Proc. of International Workshop on Future Mobile and Ubiquitous Information Technologies, May 2006, pp. 221-225.

[8] T. Yamashita, K. Izumi, and K. Kurumatani, "Effect of car navigation with route information sharing on improvement of traffic efficiency," Proc. of ITSC 2004, September 2004, pp. 465-470.

[9] S. Kawakami and H. Matsui, "Transportation engineering," Morikita, 2004. (in Japanese)

[10] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamic model of traffic congestion and numerical simulation," Phys. Rev. E51, 1995, pp. 1035-1042.

[11] M. Kimura, S. Inoue, Y. Taoda, T. Dohi, and Y. Kakuda,"A novel method based on VANET for alleviating traffic congestion in urban transportations," Proc. 11th International Symposium on Autonomous Decentralized System (ISADS2013), March 2013, pp. 131-137.

 ISBN: 978-1-61208-318-6

Solving the Virtual Machine Placement Problem as a Multiple Multidimensional Knapsack Problem

Ricardo Stegh Camati, Alcides Calsavara, Luiz Lima Jr.
Programa de Pós-Graduação em Informática – PPGIa
Pontifícia Universidade Católica do Paraná – PUCPR
Curitiba, Brazil
rcamati@ppgia.pucpr.br, alcides@ppgia.pucpr.br, laplimpa@ppgia.pucpr.br

Abstract—Effective placement of virtual machines in a cluster of physical machines is essential for optimizing the use of computational resources and reducing the probability of virtual machine reallocation. Many of previous works treat virtual machine placement as an instance of the bin packing problem, as they aim at saving energy. Alternatively, we propose an approach based on the multiple multidimensional knapsack problem, where the main concern is to maximize placement ratio. Several traditionally employed placement algorithms were re-implemented and new algorithms were defined by using such an approach. The algorithms were evaluated with respect to placement ratio, by employing a novel evaluation method that contemplates variation of computational resources heterogeneity in multiple dimensions and also variation of placement density. The experimental results showed that heterogeneity of resources among physical machines impairs the placement ratio, while heterogeneity of resources among virtual machines benefits it. It was also possible to observe that increase of density placement up to a certain point benefits the placement ratio.

Keywords- cloud computing; virtual machine placement; knapsack problem; evaluation method;

I. INTRODUCTION

The client-server model widely used in modern computing implies that the demand for computer resources (e.g., processing power, memory, storage, and network bandwidth) in a given application can vary significantly over time; the demand may increase or diminish depending on certain days and determined schedules [1]. The cloud computing computation model emerged as a solution to this problem, by permitting *elasticity* of computer resources [2] [10]. Such elasticity is typically implemented through *hardware virtualization* [3], a technique known since the 1960s decade. It appeared with the *time-shared* operating systems [4] [9] and its aim was to partition hardware to isolate users and applications in a *mainframe*. During the 1980s and 1990s decades, hardware virtualization was practically forgotten despite the fact that virtualization was used in other abstractions such as Java Virtual Machine (JVM). More recently, the original concept of virtualization has been employed again, not only to isolate applications and users, but also to permit data centers (large clusters of hosts) to achieve a more dynamic and rapid deployment [5]. Thus, the basic service provided by a data center is the instantiation of virtual machines required by its clients.

There are many benefits for both data center provider and client. In general, a provider is able to sell computer resources to a large number of clients. In addition, the amount of computer resources sold can be higher than the amount actually available, by assuming that, at any given time, the amount of resources actually required by clients is lower than the contracted amount. On the other hand, a client is able to buy computer resources for lower costs, when compared to the costs of keeping a private infrastructure (hardware, software, and personnel). Moreover, since the demands for computer resources may vary over time, a client should pay just for what is actually used, thus reducing costs even further.

The intrinsic complexity of the cloud computing computation model poses a difficult problem regarding resource management. Basically, a data center provider aims at maximizing profits. That implies reducing the deployed computer resources as much as possible, i.e., within each time slot, the amount of computer resources available to clients should be the minimum necessary, according to the corresponding expected demand. However, it should be noticed that the provider must honor the contracts established with clients, whereby quality-of-service requirements, including performance requirements, are specified. In other words, the so-called Service-Level Agreement (SLA) must be fulfilled. As a rule of thumb, the number of hosts (physical machines) in a data center should be augmented gradually, as more client contracts are established. Such a measure for cost reduction is further improved by keeping active only a minimal subset of hosts that fulfills the demand for resources within each time slot, mainly to save energy. In the literature, this problem is referred to as *virtual machine consolidation*.

The critical issues in the design of a virtual machine consolidation mechanism include the choice of algorithms for virtual machine placement and migration. A virtual machine placement is the action of instantiating it as required by a client in a properly chosen host, while a virtual machine migration is the action of moving a virtual machine from one host to another. The reason for migration is that the original host becomes either overloaded or under loaded, according to some criteria. A virtual machine is moved away from an overloaded host to guarantee that computer resource demands and performance requirements are accomplished. On the other hand, it is moved away from an under loaded

 ISBN: 978-1-61208-318-6

host to be able to deactivate such host, thus saving energy. However, migration itself can introduce both some performance degradation and some extra energy consumption, so it should be employed carefully and avoided as much as possible. Unfortunately, virtual machine migration cannot be fully prevented because they are intrinsically dynamic with respect to computer resource demand, a property known as *elasticity*. Nevertheless, a proper initial placement can help to minimize the probability of host overload, consequently reducing migration.

A virtual machine placement algorithm should provide a trade-off between energy savings and overall system performance. If a placement algorithm's objective is solely energy savings, it is likely to happen that hosts become overloaded more often, and clients may experience some performance degradation caused by virtual machine migration. In spite of that, the typical approach employed in real-world data centers is based on the *Bin Packing Problem* and a corresponding rather simple solution: the *First Fit Algorithm*. Within this approach, only the minimal necessary hosts are kept active at all times and only one is candidate (normally, the least occupied host) for virtual machine placement. Some proposals [8] [11] try to improve that solution by reordering the virtual machine request queue according to some criteria before their actual placement; those are referred to as *First Fit Decreasing Algorithms*. Other proposals [16] [17] [18] [19] are based on the *Multiple Knapsack Problem* and a corresponding *Best Fit Algorithm.* Within this approach, hosts are less occupied in average and several of them are the candidates for virtual machine placement at once. As a consequence, less migration is expected at the cost of some extra energy consumption. Also, a new issue emerges regarding the number of computer resource types considered in making a choice between the candidates. Most solutions consider just one resource type, typically processing power. Other solutions consider a combination of resource types, such as processing power, memory, storage, network bandwidth, and so forth. For simplicity, each resource type is referred to as a machine dimension, so such a solution is based on the *Multidimensional Multiple Knapsack Problem*.

Some characteristics of real-world data centers can make the virtual machine placement problem even more complex. One such characteristic is machine heterogeneity, i.e., machines present different amounts for a given dimension. Another characteristic is the average density of virtual machines per host. Such characteristics may profoundly impact on the behaviour of placement algorithms, and that can be very hard to measure.

In this paper, the multidimensional multiple knapsack approach to the virtual machine placement problem is explored in several ways. Firstly, two variants of the First Fit Decreasing Algorithm that are normally employed in the bin packing approach, namely the Dot Product Algorithm and Volume Algorithm, were adapted to the multiple knapsack approach. Secondly, two novel algorithms are proposed, namely the Best Dimension Algorithm and the Osmosis Algorithm. Thirdly, a novel evaluation method for virtual machine placement algorithms is proposed in order to accomplish multiple machine dimensions, heterogeneity of machines and distinct densities of virtual machines per host.

The remaining of this paper is organized as follows. Section II discusses the virtual machine placement problem in deep. Section III describes the normally employed virtual machine placement algorithms. Section IV presents the novel evaluation method. Section V presents some experimental results. Finally, Section VI makes some concluding remarks.

II. PLACEMENT PROBLEM

Given a data center composed of a set of hosts (physical machines) and a corresponding queue of client requests for virtual machine instantiation, the virtual machine placement problem consists in determining a host of the data center to place (actually, instantiate) each virtual machine in the queue, aiming at least at one of the following objectives:

i. To minimize the electric energy consumed by the data center [13] [14]. According to [7], the costs of powering and cooling accounts for 53% of the total operational expenditure of datacenters;
ii. To maximize the placement ratio, i.e., the quotient between the number of placed virtual machines and the total number of requests in the queue;
iii. To minimize the overall systems performance degradation caused by virtual machine migration [6][20] that is triggered by *elasticity*.

The virtual machine placement problem encompasses several issues, as detailed in the sequel.

A. Machine Dimension

A typical virtual machine requires resources of different types, such as processing power (usually measured in MIPS), memory, storage, and network bandwidth [19]. Each resource type is referred to as a *machine dimension*, or simply *dimension*, since it applies to both virtual machine and host. The demand in a certain dimension can vary over time within each virtual machine, and it not known a priori. For simplicity, it is assumed that a demand initially defined is actually an upper bound. Naturally, a virtual machine must be placed in a host that can provide enough resources in all dimensions. However, the criterion to choose a host where to place a virtual machine can be based on either a single dimension or a combination of any number of dimensions. For instance, one may decide according to processing power only, such that a virtual machine is placed in the host where the processing power dimension is mostly available, as long as the resources available in the other dimensions are enough too. As another example, one may make a placement decision based on some ranking which is determined according to the available resources in the above four dimensions. While a single-dimension criterion is simple to implement, a multi-dimension criterion may help to increase the balance between resource usages, and placement ratio as well.

 ISBN: 978-1-61208-318-6

B. Delivery Mode

Although requests for virtual machine placement arrive continuously at a data center, their corresponding delivery, that is, their actual placement on hosts can be performed in two distinct modes: *(i) single delivery mode*: the first virtual machine in the queue is popped out and placed on a host, that is, virtual machines are necessarily delivered in the same order of arrival of requests; *(ii) group delivery mode*: a number of virtual machine requests in the beginning of the queue are popped out and placed on a host. That allows reordering the group of virtual machines before placement. The single delivery mode is simple to implement, it respects requests arrival order, and it causes no additional delay in virtual machine placement delivery. The group delivery mode, on the other hand, has a potential to increase the virtual machine placement ratio since virtual machine requests can be reordered in an attempt to optimize resource usage.

C. Heterogeneity

The set of hosts in a data center can vary with respect to the resources they provide. In the same fashion, the set of virtual machines to place can vary with respect to the resources they require. In other words, both set of hosts and set of virtual machines are heterogeneous with respect to machine dimensions. Such heterogeneity adds complexity to the virtual machine placement problem. On the other hand, it can be exploited to reach the defined objectives.

D. Virtual Machine Life Cycle

Once a virtual machine is instantiated (placed on an initial host), it can change its resource demands over time (*elasticity)*, it can migrate from one host to another, and it terminates eventually. Such behavior may affect the system's performance, energy consumption and resource availability. Hence, it has to be considered during virtual machine placement. It should be noticed that virtual machines are not pre-allocated, i.e., each instantiation request implies creating a new virtual machine.

III. PLACEMENT ALGORITHMS

In this section, traditionally employed virtual machine placement algorithms are explained, along with two new such algorithms are proposed, namely the *Best Dimension Algorithm* and the *Osmosis Algorithm*.

A. First Fit Algorithm

The classic *Bin Packing Problem* [11] is often adapted to the context of virtual machine placement in the following way: a whole set of virtual machines (objects) of different sizes should be placed into a series of hosts (bins) such that the minimum number of hosts are employed to place all virtual machines, thus saving energy. The First Fit Algorithm [8][11] accomplishes that by activating a single host at a time as they get filled up with virtual machines. Each virtual machine is placed in the *first* host where it *fits*, according to a predefined order between active hosts. In addition, just one virtual machine – the first one in the queue – is taken at a time, and then placed on a host. For that reason, the First Fit Algorithm is suitable for the single delivery mode.

B. First Fit Decreasing Algorithm

When the group delivery mode (as described in Section II.B) is employed, the First Fit Decreasing Algorithm [8][11] – a variant of the First Fit Algorithm – can reduce even further the average number of active hosts. The primary difference is that the set of virtual machines in the queue is ordered according to some criteria before placement. In a simple model, where just one dimension is considered (typically, CPU usage), virtual machines are ordered according to their demand for the corresponding resource. In a more sophisticated model, several dimensions can be considered to establish a ranking amongst the virtual machines. One way to defining the rank of each virtual machine is to employ the *volume method* whereby the volume of a virtual machine is calculated by multiplying its demands in all dimensions.

In a different approach, the rank of each virtual machine can be determined with respect to a reference host witch is, normally, the least occupied active host or the last one to be activated. In this approach, at least during rank calculation, the virtual machine placement problem can be modeled as an instance of the *0-1, or Binary, Knapsack Problem* [12], as follows. Given a set (group) of virtual machines (objects) where each virtual machine has an associated value, a subset must be selected and associated to a single host (knapsack) such that the *profit* is maximized. An example of an algorithm that takes this approach is one that employs the *dot product* method. The *dot product* of a virtual machine is calculated as the sum of a series of products, where each product corresponds to a dimension, and is obtained by multiplying the virtual machine demand by the reference host availability in that dimension.

For simplicity, the First Fit Decreasing Algorithm based on the volume method is referred to as *Volume Algorithm* [15], while the First Fit Decreasing Algorithm based on the dot product method is referred to as *Dot Product Algorithm* [8].

C. Best Fit Algorithm

Besides saving energy, a data center should to take measures to fulfill all Service-Level Agreement (SLA) requirements, including virtual machine placement ratio (which is equivalent to establishing a maximum time for virtual machine placement), virtual machine performance and virtual machine elasticity.

The Best Fit Algorithm keeps active a set of hosts at all times (instead of activating a single host at time) and places each virtual machine where it best fits to achieve load balance amongst the active hosts. A proper load balance should increase the probability of success in virtual machine

 ISBN: 978-1-61208-318-6

placements. Moreover, it should increase the probability of having enough resources at a given host to fulfill the needs due to a virtual machine expansion. As a consequence, virtual machine migration should be reduced, thus preventing systems performance degradation.

Basically, the problem solved by the Best Fit Algorithm can be seen as an instance of the *Multiple Knapsack Problem*: given an initial set of virtual machines (objects) of different sizes and values, a subset of it should be selected, and then placed into a set of hosts (knapsacks) such that the aggregate value is maximized. Moreover, when multiple dimensions are considered, the problem is an instance of the *Multiple Multidimensional Knapsack Problem* [16][17][18][19]. Particularly, if all virtual machines are assumed to hold a unique value, the Best Fit Algorithm maximizes the quantity of virtual machines that are placed. The remaining issue concerns the delivery mode, as discussed in the sequel.

1) Single delivery mode

Since just one virtual machine – the first one in the queue – is taken at a time, and then placed on a host, there is no actual virtual machine selection in the single delivery mode. Such a reference virtual machine will be simply either placed or discarded (actually, it can be reinserted at the end of the queue), depending on the resource availability on the active hosts. Nevertheless, the analogy with the Multiple Knapsack Problem holds if the whole queue of virtual machines (built within a certain period of time) is considered as the initial set of objects. In this case, the selected virtual machines are the ones that are placed, i.e., the ones that are not discarded. Thus, the purpose of the Best Fit Algorithm is to discard virtual machines the least as possible. In other words, the purpose is to maximize placement ratio in the long run. The actual issue, hence, is to determine the host, if there is any, where the reference virtual machine fits best. That requires ordering the set of active hosts according to some criteria, before placing the reference virtual machine. Similar to the First Fit Decreasing Algorithm (described in Section III.B), both *volume* and *dot product* methods can be employed in this case. If the volume technique is employed, host are sorted in decreasing order by their *free volume*, which is calculated as the product of available resource in each dimension. If the dot product is employed, the rank of each host is calculated as the sum of a series of products, where each product corresponds to a dimension, and is obtained by multiplying the reference virtual machine demand by the host availability in that dimension. A slightly different version of the dot product method is the *best dimension* method, proposed here, by which the product for each dimension is calculated in the same fashion as in the dot product method, but the rank of each host is set simply as the highest product. For simplicity, the Best Fit Algorithm based on the best dimension method is referred to as *Best Dimension Algorithm*.

2) Group delivery mode

The analogy with the Multiple Knapsack Problem is direct in the case of the group delivery mode, since it permits a selection of the virtual machines that best fit the set of active hosts, before placement. Actually, there are two kinds of selection that can be exploited in any form by a placement algorithm: a selection of virtual machines and a selection of active hosts.

An algorithm that employs only a virtual machine selection method should order the set of virtual machines according to some criteria, and then place each virtual machine in any active host. For example, an algorithm may order the virtual machines according to their required processing power. If more dimensions should be considered, they can be ordered according to the required *volume*.

An algorithm that employs only a host selection method should order the set of active hosts according to some criteria before placing each virtual machine that is simply popped out from the queue. For example, the set of active hosts can be ordered according to the proximity between their level of free processing power and the average processing power required by the all virtual machines in the group. It should be noticed that such an algorithm differs from an algorithm employed for the single delivery mode because its host selection method uses attribute values of all virtual machines in the group, instead of attribute values of just the first virtual machine in the queue.

An algorithm that employs both selection methods simultaneously actually employs a *match method* between virtual machines and active hosts. For example, a simple match method is to associate virtual machines that require higher processing power with hosts with higher level of free processing power.

The *Osmosis Algorithm*, proposed here, employs the host selection method only, albeit it can be easily extended to employ the virtual machine selection method as well is straightforward. The Osmosis Algorithm attempts to preserve resources that are scarce in the data center with respect to the current demand. Its host selection method consists in ordering the set of active hosts according to their availability of relatively least available resources. Hosts that hold resources whose usage levels are less critical in the data center are used first. The first step is to determine which resources present more critical levels of usage. That is achieved by determining the *weight* of each dimension as the quotient between its total demand and its total availability. The *total demand* of a dimension is defined by the sum of all demands required the virtual machines in the group, while the *total availability* of a dimension is the sum of free resource in all the active hosts. Once the weight of each dimension has been determined, the rank of each active host must be calculated with respect to the first virtual machine in the queue – the reference virtual machine – as follows. For each host, the *weighted offer ratio* of a dimension is defined as the product between the weight of that dimension and the corresponding *offer ratio,* which is

 ISBN: 978-1-61208-318-6

defined as the quotient between the host availability and the reference virtual machine demand for that dimension. Finally, the rank of each host is calculated as the sum of its weighted offer ratios, in all dimensions.

D. Discussion

The First Fit Algorithm and its variants make it possible for a data center to save energy, since only hosts that contain some virtual machine need to be on. However, because host resource usage levels tend to be close to the maximum, virtual machine migration is more likely to happen in the presence of elasticity (when, within an already running virtual machine, the demand for a given resource increases), thus degrading performance, besides consuming some extra energy.

On the other hand, the Best Fit Algorithm and its variants should improve placement ratio, which brings benefits for both data center provider and client, thus justifying the extra energy cost. In addition, virtual machine migration caused by host overload is less likely to happen, hence improving the overall systems performance.

In the following sections, an evaluation of the different placement algorithms, with respect to placement ratio, is presented, while evaluation of issues regarding elasticity, migration, performance and energy consumption are left as future work.

IV. EVALUATION METHOD

A new method to evaluate virtual machine placement algorithms is proposed here in order to take into account multidimensional machines, and to investigate the impact of machine heterogeneity and virtual machine density on placement ratio. Basically, the method consists in determining a set of virtual machines to place on a set of hosts, and applying several placement algorithms in order to compare their behavior with respect to placement ratio. Every machine (either host or virtual machine) is assumed to have a number of dimensions, where each dimension corresponds to a certain type of resource. The set of hosts is assumed to be heterogeneous with respect to the resource capacity and, for each dimension, each host may have a different capacity. In the same way, the set of virtual machines should present a certain degree of heterogeneity.

The virtual machine placement problem can be formalized as follows.

Be:

- $H = \{h_1,...,h_m\}$: *a set of m hosts*
- $V = \{v_1,...,v_n\}$: *a set of n virtual machines*
- $D = \{d_1,...,d_g\}$: *a set of g dimensions*
- T_j : *the arrival time of* $v_j \in V$ *such that* $\forall v_i, v_j : \; i < j \;\Leftrightarrow\; T_i < T_j$
- $w_{j,k}$: *the demand of* $v_j \in V$ *in dimension k*
- $c_{i,k}$: *the total capacity of* $h_i \in H$ *in dimension k*
- $L_{i,k}(T_j)$: *the free capacity of dimension k in* $h_i \in H$ *at* T_j
- $x_{i,j} \in \{0,1\}, \quad 1 \le i \le m, \quad 1 \le j \le n$
  - $x_{i,j} = 1 \Leftrightarrow v_j \in V$ *is placed in* $h_i \in H$
  - $x_{i,j} = 1 \Rightarrow x_{k,j} = 0, \forall k \ne i$

Maximize $\sum_{i=1}^{m}\sum_{j=1}^{n} x_{i,j}$

Subject to:

(i) $\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{g} x_{i,j}.w_{j,k} \le c_{i,j}$

(ii) $\forall v_j \in V, \;\; \exists \;\; S \subseteq H, \;\; S \ne \varnothing \;\; | \;\; \forall h_i \in S, \;\; \forall d_k \in D, \;\; L_{i,k}(T_j) \;\; \ge \;\; w_{j,k} \;\; \Rightarrow \;\; x_{t,j} = 1, \;\; h_t \in s$

The analysis of the effects of heterogeneity on placement ratio employs an *ideal scenario* as a starting point. In such a scenario, hosts and virtual machines are assumed to be homogeneous, and their capacities are such that all virtual machines are placed and, also, no resource is left in any host. For a given set of hosts, the number of virtual machines is determined by according to an arbitrarily fixed virtual machine density; if distinct values of virtual machine density are employed, then the number of virtual machines is determined accordingly. In the same fashion, for each machine dimension, the corresponding capacity in each host and the demand of each virtual machine are determined according to virtual machine density. Hence, the ideal scenario for a given set of hosts and a given virtual machine density includes the number of virtual machines to place, and the capacity of hosts and the demand of virtual machines for each machine dimension.

After having established an ideal scenario, other scenarios can be devised by introducing some degree of heterogeneity in both set of hosts and set of virtual machines. In such a scenario, for any given dimension and a set of machines, there is a specific range that includes all corresponding dimension values. A range can be defined by a minimum, a maximum and a corresponding median that is determined according to the ideal scenario (host capacity or virtual machine demand). The relative variation of dimension values with respect to the median is called the

 ISBN: 978-1-61208-318-6

amplitude of that dimension with respect to that set of machines.

Formally, given a range *[minimum, maximum]* of values of a dimension *d* in a set of *M* of machines, the *amplitude* of *d* with respect to *M* – denoted as $A_{M,d}$ – is defined in (1).

$$A_{M,d} = 100\ (maximum-median)/median \quad (1)$$

In a set of *M* of machines where each machine has *n* dimensions that are labeled from *1* to *n*, the tuple containing the amplitudes for all dimensions – denoted as $A^*_{M,d}$ – is defined in (2).

$$A^*_{M,d} = (A_{M,1},...,A_{M,n}) \quad (2)$$

Given a set *V* of virtual machines that should be placed, let $P \subseteq V$ be the set of virtual machines actually placed after applying a certain algorithm, *v* be the number of machines in *V*, and *p* be the number of machines in *P*. The corresponding *placement ratio* -- denoted as τ -- is defined in (3).

$$\tau = \frac{p}{v} \quad (3)$$

The *ideal scenario* is one such that $\tau = 1$, i.e., the whole set of virtual machines is successfully placed, and also all host resources are fully occupied. Such a scenario can be easily synthesized when there are *v* virtual machines to place on *h* hosts, by assuming that:

(i) All amplitude values are *zero*, for both hosts and virtual machines. In other words, hosts and virtual machines are homogeneous.

(ii) For each machine dimension, let *k* be the corresponding capacity in each host, and *t* be the corresponding demand of each virtual machine. Then, $k \bmod t = 0$ (there is no resource left in each host) and $h \cdot k = v \cdot t$ (there is no resource left in the data center).

Given a set of *h* hosts and a set of *v* virtual machines, the average number of virtual machines that should be placed per host – denoted as ρ – is simply defined in (4).

$$\rho = \frac{v}{h} \quad (4)$$

In the ideal scenario, $\rho = \frac{k}{t}$ for any machine dimension.

V. EXPERIMENTS

The algorithms described in Section III were experimentally evaluated with respect to placement ratio, according to the method described in Section IV. The experiments were based on simulation by employing a custom simulator (available for download at http://www.ppgia.pucpr.br/~alcides/PSim) written in Python language, and they were divided into seven scenarios that correspond to distinct degrees of machine heterogeneity, and distinct densities of virtual machines, as well. The number of hosts is fixed to 100 for all scenarios, except for Scenario VI, where this number is ten due to simulation time constraints. For most scenarios, virtual machine density is fixed to ten virtual machines per host, which is a typical density for small instances in data centers. The number of dimensions is fixed to four since the most commonly considered resources are processing power, memory, storage, and network bandwidth. Consequently, the amplitude tuples contain four values corresponding to heterogeneity of those resources. For simplicity, in the experimental results shown above, a host amplitude tuple is denoted as δ, while a virtual machine amplitude tuple is denoted as π.

In scenarios I and II, host amplitude is fixed in order to analyze the impact of the virtual machine amplitude variation, while, in scenarios III and IV, virtual machine amplitude is fixed in order to analyze the impact of the host amplitude variation. Also, in scenarios I and III, the machine amplitude variation increases evenly, while it increases non-uniformly in scenarios II and IV.

According to Fig. 1 and Fig. 2, increasing virtual machine amplitude either evenly or non-uniformly favors all algorithms. The Dot Product Algorithm was the best performer, achieving a placement ratio from 2% to 10% better than the First Fit Algorithm.

According to Fig. 3 and Fig. 4, increasing the host amplitude variation either evenly or non-uniformly impairs the performance of all algorithms. The Dot Product Algorithm shows the best overall performance, achieving from 6% to 19% more performance than the First Fit Algorithm.

In scenarios V and VI, the impact of virtual machine density is analyzed. In this case, host amplitude and virtual machine amplitude are fixed. While Scenario V considers virtual machines that have a monolithic operation system as guest, the Scenario VI considers very small virtual machines that have a microkernel operational system as guest [21]. The results shown in Fig. 5 are quite interesting. Increasing the virtual machine density favors all algorithms. The Osmosis Algorithm was the best solution for very small virtual machine density (from 2 to 3), performing from 5% to 9% better than the Volume Algorithm, the worst performer. With a density higher than 3, once more the Dot Product Algorithm was the best performer, achieving a placement ratio from 3% to 12% higher than the First Fist, the worst performer in this range.

 ISBN: 978-1-61208-318-6

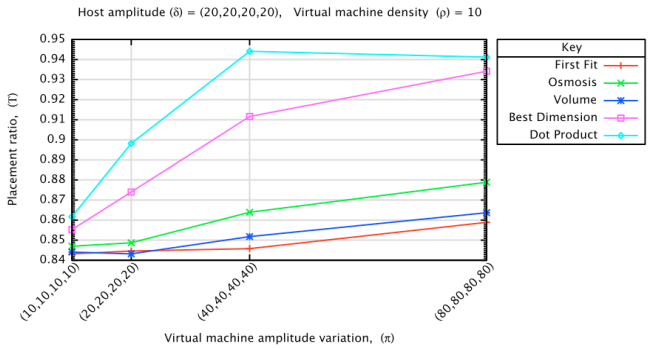


Figure 1. Increasing virtual machine amplitude variation evenly.

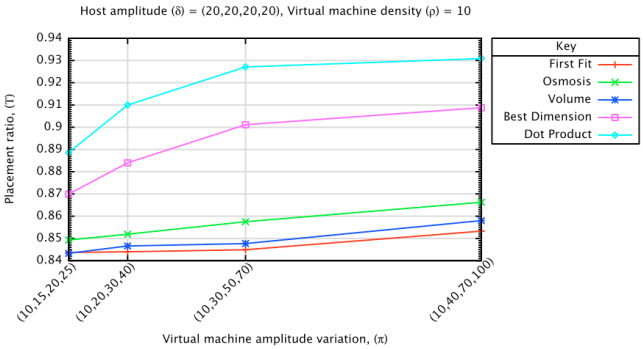


Figure 2. Increasing virtual machine amplitude variation non-uniformly.

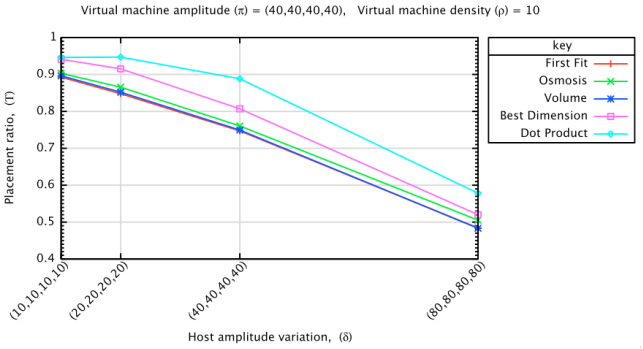


Figure 3. Increasing host amplitude variation evenly.

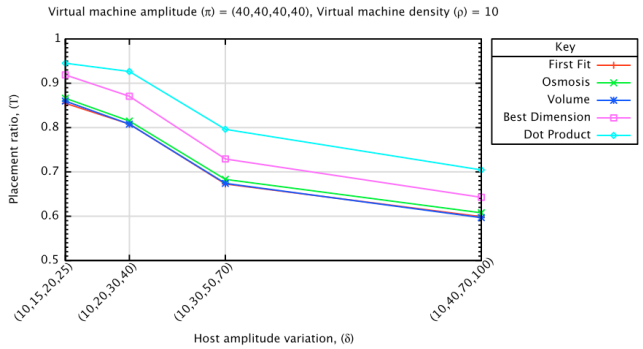


Figure 4. Increasing host amplitude variation non-uniformly.

According to Fig. 6, increasing density is beneficial for all algorithms, at least up to a certain point. As expected, the Dot Product Algorithm proved to be superior to all other algorithms, being from 6% to 9% more efficient than the First Fit Algorithm and the Volume Algorithm.

In Scenario VII, placement ration of the Dot Product Algorithm is analyzed by varying both host amplitude and virtual machine amplitude, simultaneously. The choice of the algorithm is due to its higher performance, as verified in previous scenarios. In this scenario, virtual machine density is fixed as 10, as well. In Fig. 7, it can be noticed there are several depressions caused by increasing host amplitude variation and also by the decreasing virtual machine amplitude variation. It can be noticed that virtual machine heterogeneity tends to improve placement ratio, while host heterogeneity tends to degrade it.

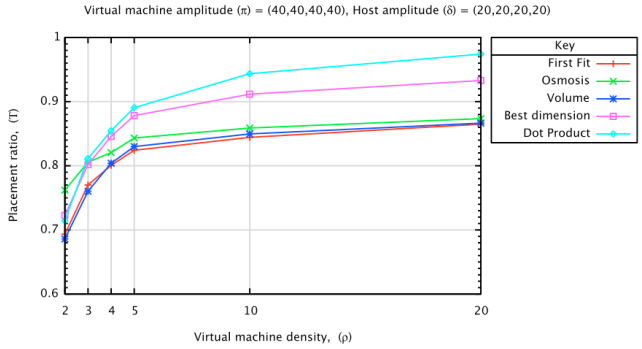


Figure 5. Increasing virtual machine density in a monolithic approach.

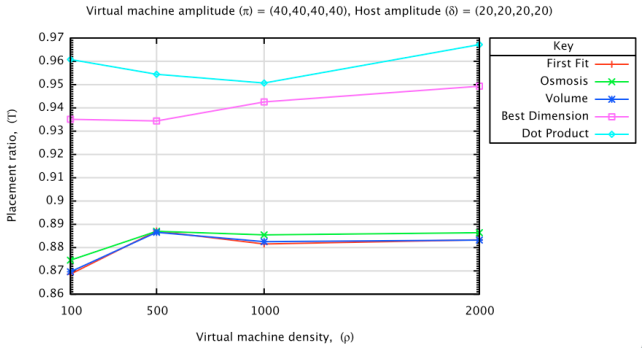


Figure 6. Increasing virtual machine density in a microkernel approach.

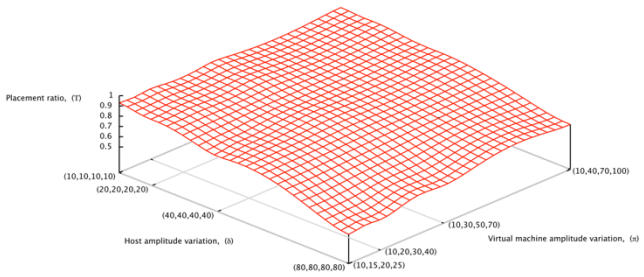


Figure 7. Increasing both host amplitude variation evenly and virtual machine amplitude variation non-uniformly

 ISBN: 978-1-61208-318-6

In conclusion, the Dot Product Algorithm seems to be the best solution in most cases. The Best Dimension Algorithm and the Osmosis Algorithm, proposed in this paper, showed a median performance. While Best Dimension has a better overall performance than Osmosis, Osmosis is the best choice in the case of very low virtual machine density, surpassing even the Dot Product Algorithm. A real world scenario with 2 or 3 virtual machines per host is a very common with large instances, as an example a LAMP (Linux, Apache, MySQL, PHP) Application. The First Fit Algorithm and the Volume Algorithm are very low performers. On the other hand, as they are simple to implement, they can be appropriate when there is a performance bottleneck in the virtual machine placement system.

VI. CONCLUSION

This work has shown that the multidimensional multiple knapsack approach can be more effective than the bin packing approach to the virtual machine placement problem; in general, the First Fit Algorithm was the worst performer in our experiments. The proposed evaluation method permitted to represent critical real-world data center characteristic, and it proved to be easy to use. The experiments have shown that virtual machine heterogeneity plays a favorable role in virtual machine placement, while host heterogeneity has the opposite effect. Also, in general, high virtual machine density favors virtual machine placement ratio. Finally, algorithms that use properties of virtual machines (Dot Product, Best Dimension and Osmosis Algorithm) tend to have a better performance.

The proposed evaluation method can be extended to include energy consumption, elasticity, migration, and termination. More future works include the evaluation of other types (genetic, ant colony, etc.) of placement algorithms, and experimentation with more fixed values for virtual machine density. Also, the costs of running each placement algorithm should be computed to verify its feasibility in large-scale scenarios, and to verify if the corresponding gain in placement ratio is worthwhile. Other issues regarding SLA, such as services response time and security are left as future work as well.

REFERENCES

[1] H. Liu and S. Wee, “Web Server Famrm in Cloud: Performance Evaluation and Dynamic Architecture”, Proceedings of the First International Conference on Cloud Computing, Technology and Science, Berlin and Heidelberg, Springer Verlag, 2009, pp. 369-380.

[2] P. Mell and T. Grance, “The NIST Definition of Cloud Computing”, Technical Report, Gaithersburg, 2011, National Institute of Technology Special Publication 800 – 145.

[3] E. J. Smith and R. Nair, “The Architecture of Virtual Machines”, Computer, vol.38(5), May. 2005, pp. 32-38.

[4] V. Melinda, “VM and the VM Community: Past, Present and Future”, Office of Computing and Information Technology, Princeton University, Share 89 Sessions 9059-9061, 1997, pp. 1-68.

[5] M. Rosenblum, “The Reincarnation of Virtual Machines. Virtual Machines”, Virtual Machines, vol.2(5), Aug. 2004, pp. 34-40.

[6] M. R. Hinnes, U. Deshpande, and K. Gopalan, “Post Copy Live Migration of Virtual Machines”, ACM SIGOPS Operating System Review, vol.43(3), 2009, pp. 14-26.

[7] Q. Zhang, L. Cheng, and R. Boutaba, “Cloud Computing: State of Art and Research Challenges”, Journal of Internet Services and Applications,vol. 1(1), Apr. 2010, pp. 17-18.

[8] R. Panigrahy, K. Talwar, L. Uyeda, and U. Wider, “Heuristics for Vector Bin Packing”, Microsoft’s VMM Product Group, Microsoft Research Sillicon Valley, 2011.

[9] R. J. Creasy, “The Origin of VM/370 Time-Sharing System”, IBM Journal of Research and Development, vol. 25(5), pp. 483-490.

[10] T. P. Endo, E. G. Gonçalves, J. Kelner, and H. D. F. Sadok, “A Survey on Open-Source Cloud Computing Solutions”, Proceedings of the XXVII Brazilian Simposium of Computer Networks and Distribuited Systems, VII Workshop in Cloud Computing, Porto Alegre, Brazil, 2010, pp. 3-16.

[11] B. Xia and Z. Tan, “Tigther bounds of the First Fit Algorithm for the Bin-Packing Problem”, Elsevier, Hangzhou, vol. 158(15), Aug. 2010, pp. 1668-1675.

[12] D. Pisinger, “Algorithms for Knapsack Problems”, Department of Computer Science of University of Copenhagen, Copenhagen, Feb. 1995.

[13] Y. Wu, M. Tang, and W. Fraser, “A Simulated Annealing Algorithm for Energy Efficient Virtual Machine Placement”, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Seoul, Oct. 2012, pp. 14-17.

[14] J. Xu and J. A. B. Fortes, “Multi-Objective Virtual Machine Placement in Virtualized Data Center Enviroments”, Proceedings of the 2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Pysical and Social Computing, Washington, 2010, pp. 179-188.

[15] D. Bonde, “Techniques for Virtual Machine Placement in Clouds”, Departament of Computer Science and Engineering - MPT Stage I Report on the Indian Institute of Technology, Bombay, 2010, ROLL No:08305910.

[16] S. Fidanova, “Heuristics for Multiple Knapsacks Problem”, Proceeding of the IADIS International Conference on Applied Computing, Algarve, 2005, pp. 225-260.

[17] Y. Song, C. Zhang, and Y. Fang, “Multiple Multidimensional Knapsack Problem and it’s Applications in Cognitive Radio Networks”, Military Communications Conference, San Diego, Nov. 2008, pp. 1-7.

[18] A. Singh, M. Korupolu, and D. Mohapata, Server-Storage Virtualization: Integration and Load Balance in Data Centers, International Conference for High performance Computing, Network, Storage and Analyses, pp. 1-12.

[19] E. Mohamadi, M. Karimi, and S. R. Heikalabad, “A Novel Virtual Placement in Virtual Computing”, Australian Journal of Basic and Applied Scienes, Australia, vol. 5(10), 2011, pp. 1149-1555.

[20] D. Magalhães, J. M. Soares, and D. G. Gomes, “Virtual Machine Migration Impact Analysis in a Virtualized Computer Enviroment”, Proceedings of the XXIX Brazilian Symposium on Computer Networks and Distributed Systems, Campo Grande, 2011, pp. 235-248.

[21] A. Whiteaker, M. Shaw, and S. T. Gribble, “Denali: A Scalable Isolation Kernel”, Proceedings of the 10th Workshop on ACM SIGOPS European Workshop, New York, 2002, pp. 10-15.

 ISBN: 978-1-61208-318-6

Comparing Network Traffic Probes based on Commodity Hardware

Luis Zabala[(1)(2)], Alberto Pineda[(1)], Armando Ferro[(1)], Daniel Fernández[(2)]
[(1)]Department of Communications Engineering, University of the Basque Country (UPV/EHU), Bilbao, Spain
[(2)]Stochastic and Operations Research – Networks (NET), Basque Center for Applied Mathematics (BCAM), Bilbao, Spain

Emails: {luis.zabala, alberto.pineda, armando.ferro}@ehu.es
Emails: {lzabala, dfernandez}@bcamath.org

***Abstract*—Due to the fact that, nowadays, it is possible to capture traffic in 1-10 Gigabit Ethernet networks using commodity hardware, many traffic monitoring systems, and especially capturing tools, have been proposed in recent years. This paper presents a comparison between two software probes named Adviser and Ksensor. Both of them are multi-processor systems and are built over conventional hardware. However, while Adviser is designed in user space, Ksensor runs in kernel space. This work compares the performance results of the two probes considering several capture engines (NAPI, PF_RING with DNA, PFQ) and, at the same time, different application or analysis loads. The evaluations of the probes with the different settings have been performed on the same hardware multi-core configuration. The results of the evaluations let conclude which solution is better in each situation and which solution must be discarded.**

Keywords-packet capturing; Ksensor; Adviser; NAPI, PF_RING; PFQ

I. INTRODUCTION

Nowadays, network traffic capturing and analyzing systems are becoming increasingly relevant. Different applications can be related to these traffic monitoring systems, for example, network antiviruses, Quality of Service monitoring, intrusion detection systems, traffic classification and balancing. They can also help administrators in network troubleshooting. As the speed of the network links increases, the performance requirements on the monitoring systems are more severe. In particular, in multi-Gigabit environments, overload situations can happen, reaching the system limitations in terms of memory occupation, CPU usage, system bus throughput, and having negative impact on the accuracy of the monitoring application. Therefore, as far as possible, unnecessary consumption of available resources must be avoided.

The packet capturing stage is an essential component of the traffic monitoring system. The evolution of commodity hardware has made possible the capture of network traffic to be a feasible task over high-speed networks, without using any neither specific nor expensive hardware [1]. This way, several research works [2][3][4] have arisen focused on the development of analysis systems that are able to process all the information carried by actual networks. Among them, our research group of the UPV/EHU, called Network, Quality and Security (NQaS), is working on software solution proposals for traffic analysis systems over multi-processor architectures.

Two traffic probes have been developed by NQaS. They are generic and flexible, and they allow doing any type of analysis on the captured traffic. Due to the fact that the monitoring application is over a multi-processor platform, the analysis can be done concurrently, obtaining a high performance. This paper presents a comparison between those two probes which have different view of design. On the one hand, Adviser is a generic multi-processor architecture which has been built in user space, it is portable and it can make use of different capturing systems. On the other hand, Ksensor is a kernel-space framework in which the processing modules have been migrated from user-level to the kernel of the operating system.

Many capturing tools and comparisons have made available in the literature. However, most of them do not asses how the packet capture is affected under different analysis or application loads. This work compares the performance results of the probes Adviser and Ksensor considering different capture engines and, at the same time, different analysis loads as will be seen below.

The rest of the paper is organized as follows. In Section II, a brief explanation about related work is introduced. In Section III, we describe the network traffic probes that will be compared later. Section IV presents the test setup for evaluating the performance of the traffic analysis systems. Section V shows the results of our measurements. Finally, Section VI remarks the conclusions.

II. RELATED WORK

The improvement of packet capturing capabilities with commodity hardware has been an extensively covered research topic. Hardware and software solutions have been proposed.

Among the most recent software solutions, it is remarkable Luca Deri's numerous contributions within the project ntop [2]. In this project, an open source platform has been developed to monitor traffic in high speed networks and it has given rise to interesting works such as [5], which presents the network socket PF_RING, [6], in which nCap, a proposal based on commercial network cards was proposed, and [7], which deals with aspects related to the packet parallel processing in multi-core platforms, as well as with the driver called Threaded New API (TNAPI), which uses a multiqueue structure. The same research group has also

 ISBN: 978-1-61208-318-6

presented the framework vPFRING for capturing packets on virtual machines running on commodity hardware [8].

The project Ringmap [3] has certain similarities with ntop, since it also proposes to improve the performance of packet capture removing some packet copy operations and mapping the Direct Memory Access (DMA) buffer into the user space. Ringmap works with FreeBSD operating system, while ntop works with Linux. Another approach proposed to speed up the packet capturing capability is Netmap [4]. This is a BSD based project which integrates in the same interface a number of modified drivers mapping the NIC transmit and receive buffers directly into user space. [9] proposes a packet capturing engine with multi-core commodity hardware named PFQ, which allows parallel packet capturing in the kernel and, at the same time, to split and balance the captured packets across a user-defined set of capturing sockets. Even, there have been various works [1][10][11] in recent years looking at evaluating existing packet capture techniques. In particular, [11] evaluates and compares different capture solutions for Linux and FreeBSD operating systems. The evaluation shows that FreeBSD outperforms standard Linux PF_PACKET, Linux with PF_RING performs better than PF_PACKET and even better than FreeBSD if multiple capturing processes are run on the system. Another option analyzed is TNAPI, which achieves the best performance when it is combined with PF_RING.

III. DESCRIPTION OF THE SOFTWARE PROBES

As mentioned before, the performance of two software probes will be compared in this paper. The first one, named Adviser, is a user-level traffic probe, i.e., it has got the common structure with the analysis or monitoring application in user space and the capturing stage in kernel space. Adviser admits different configurations for the capturing as will be explained later. The second probe is called Ksensor and it is an entirely kernel-level probe. Both of them capture and analyze traffic in Gigabit Ethernet networks.

A. *Adviser. The User-Level Framework*

Adviser [12] is a multi-processor architecture able to capture network traffic and analyze it applying online complex algorithm. Since the architecture is built on top of the operating system, it is portable to several systems. Fig. 1 shows the block diagram of Adviser framework. It works essentially as follows.

First, the system parser interprets the configuration files and stores system logic in memory. Then, analysis engine processes captured packet from the network according to the logic stored in memory. After applying the rules, the engine stores the results of the analysis. Finally, offline processing module takes these results from memory and handles this information to provide traffic statistics or reports.

There is also a module called periodic action manager, which supports dynamic activation or deactivation of rules, modification of period time, etc.

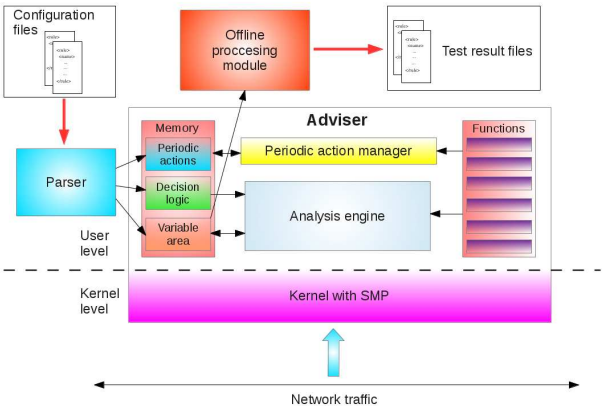


Figure 1. Adviser framework.

The traffic capturing system is in the kernel with Symmetric Multi-Processing (SMP). As Adviser can be configured with different capturing systems, in order to obtain Adviser's performance results with different configurations, we have integrated three capturing systems into Adviser, as follows.

1) Adviser's capturing system with NAPI and LibPcap

This first setting uses the network subsystem of standard GNU/Linux. It is New API (NAPI) [13] from kernel versions higher than 2.4. The link between the Linux networking subsystem and the user-space application Adviser is established by using the library LibPcap [14]. As can be observed in Fig. 2a, the application Adviser reads packets from the socket queue through Libpcap. Once Adviser's analysis engine receives the packet, it is decoded and the analysis logic is applied to it.

2) Adviser's capturing system with PF_RING

In order to reduce the number of copies from the moment that the packet arrives to the capture system until it is delivered to the application, we set out the use of PF_RING [3] as capturing system in Adviser. In this point, there are different options for doing the integration. One of them is the use of PF_RING with LibPcap and a PF_RING aware NIC driver. However, there is another one which provides a better performance and, for this reason, we select it for implementing. It is the integration of PF_RING with the driver Direct NIC Access (DNA) [2] into Adviser, which allows to map NIC memory and registers to the user space. This way, packet copy from the NIC to the DMA ring is done by the NIC Network Process Unit and not by NAPI, resulting in better line-rate captures. Fig. 2b shows Adviser with PF_RING DNA.

Some adaptation modules are needed to integrate PF_RING into Adviser. First, a new module is responsible for managing the operations of PF_RING, such as the creation of the capturing ring and the interaction with the network interface to set filtering rules or working modes.

 ISBN: 978-1-61208-318-6

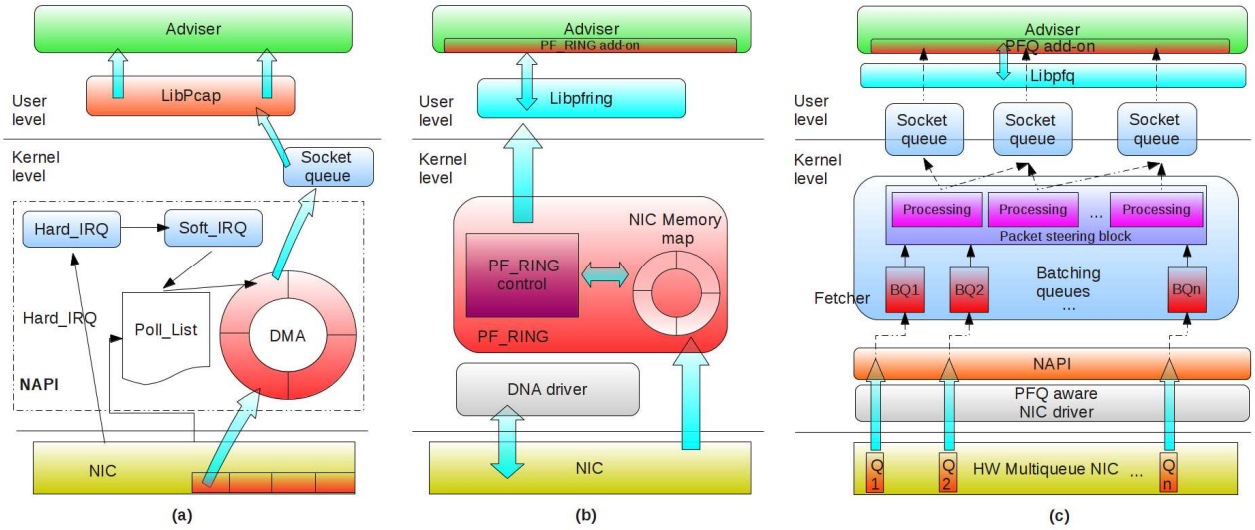


Figure 2. Adviser capturing packets (a) with NAPI and LibPcap (b) with PF_RING (c) with PFQ.

Once the capturing ring is created, a socket is enabled, the packet capturing starts and the application access to the ring through the socket. When the packet is captured, PF_RING places its contain in a data structure whose format is different from the one used by LibPcap. For this, a new module fits the format and the sizes of those data structures so that Adviser receives the data properly to be decoded.

The last adaptation is related to the concurrency system. Due to the design of PF_RING, the integration of Adviser and PF_RING has to be based on threads, instead of processes. For this reason, a new module is responsible for creating and managing threads to set an access control to the critical sections. The library Libpfring provides a control mechanism called spinlock, which allows one thread to access to the protected code, while the rest of the threads are blocked in an active-standby process.

3) Adviser's capturing system with PFQ

PFQ [15] is a network-capture engine designed for the Linux kernel 3.x and modern 64-bit architectures. It is optimized for multi-core processors, as well as for network devices supporting multiple hardware queues.

Adviser with PFQ is depicted in Fig. 2c. PFQ consists of the following components: the fetcher, the packet steering block and socket queues [9]. The fetcher dequeues the packet directly from the driver, which can be a standard driver or a patched "aware" driver, and inserts it into the batching queue. The next stage is represented by the packet steering block, which is in charge of selecting which socket needs to receive the packet. The final component is the socket queue, which represents the interface between user space and kernel space. Every kernel processing (from the reception of the packet up to its copy into the socket queue) is carried out within the NAPI context; the last processing stage is performed by Adviser at user space.

As in the case of PF_RING, an adaptation is necessary for the integration of Adviser with PFQ. To do this, using the tools provided by Libpfq [16], a PFQ add-on is created in Adviser. This access from Adviser to PFQ is based on threads.

B. *Ksensor. The Kernel-Level Framework*

Ksensor [17] is a kernel-level multi-processor monitoring system for high speed networks which uses commodity hardware. Its design (see Fig. 3) is based on the migration of the processing modules from user-level to the kernel of the operating system. Only system configuration (Parser) and result management (Offline Processing Module) modules remain at user-level.

First, the system parser interprets the configuration files and stores system logic in memory. Then, analysis engine processes captured packet from the network according to the logic stored in memory. After applying the rules, the engine stores the results of the analysis. Finally, offline processing module takes these results from memory and handles this information to provide traffic statistics or reports.

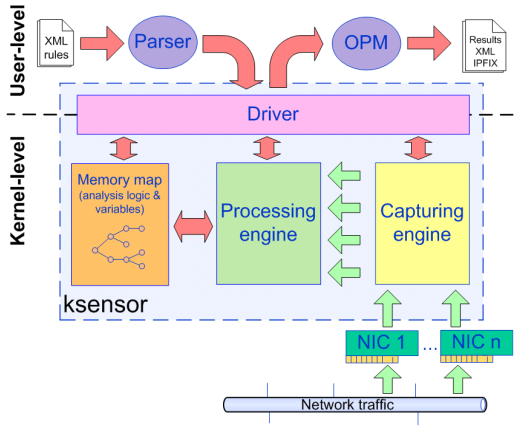


Figure 3. Ksensor framework.

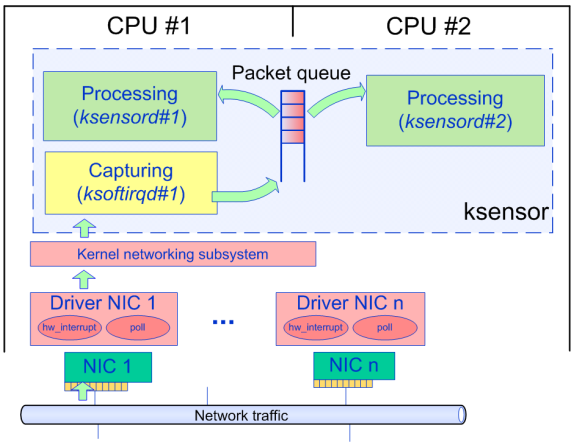


Figure 4. Execution instances in Ksensor with two processors and one NIC.

There are defined as many analyzing kernel threads (ksensord#n in Fig. 4) as the number of processors on the hardware. Each thread belongs to an execution instance of the system (capture and analysis). All threads share information through the kernel memory.

Regarding the capture, it is based on the kernel networking subsystem, i.e., NAPI. There are as many capturing instances (ksoftirqd#n in Fig. 4) as capturing NICs (IRQ affinity). A single packet queue is shared by all the analyzing instances (see Fig. 4).

In order to prevent livelock situations at high packet arrival rates, there is a congestion avoidance mechanism. It also prevents Ksensor from wasting resources in the capture of packets that the system will not be able to process later. When the packet queue reaches a maximum number of packets, this mechanism forces NAPI to stop capturing packets. This means that all the resources of all the processors are dedicated to analyzing instances. When the number of packets in the packet queue reaches a fixed threshold value the system starts capturing again.

IV. TEST SETUP FOR COMPARING THE PROBES

The tests done in order to compare the probes are very important. Firstly, in order to automate the tests, a software architecture has been designed and implemented by NQaS research group. This architecture configures the tests, runs them and gathers the results automatically. It consists of four types of logical elements: manager, agents, daemons, and formatters.

A. Software and Hardware Details

The real environment where the different probes have been tested can be seen in Fig. 5. There are two networks. One is called management network and it is used for sending the configuration commands from the manager to the agents and the statistics of the test from the agents to the manager. The other one is called capturing network and it is used for testing the probes.

The machine called manager is the interface between the testing architecture and the administrator.

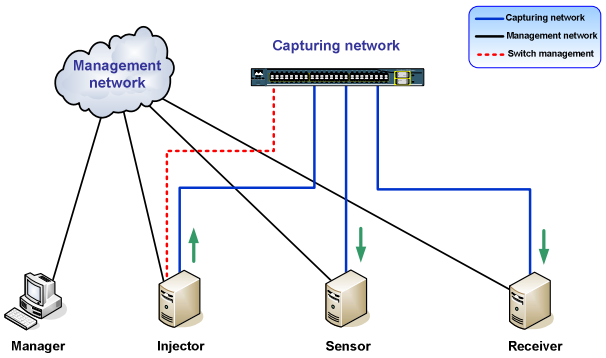


Figure 5. Network infrastructure to test the probes.

The injector is in charge of generating synthetic network traffic in order to simulate traffic load in the network. In order to do that, this machine has an Endace DAG 4.3GE card that allows injecting traffic rates up to 1 Gbps. The machine has got two processors Intel Xeon 5110 at 1.66 GHz and 2 GB of RAM memory. It runs a Debian GNU/Linux.

In the machine called Sensor run all the probes. The different implementations of Adviser are made using a Debian 7 with a kernel Linux 2.6.35. On the other hand, Ksensor is a modification of the kernel Linux 2.6.23 with a kernel module that implements the analysis tasks. The machine has got two processors Intel Quad Xeon 5420 at 2.5 GHz with 4 GB of RAM memory. Each processor has got four cores.

The receiver machine is the one that should receive the traffic. It is only used for extracting statistics.

These three machines, in order to configure the implied software and to collect the statistics, run an agent and several daemons of the testing architecture.

B. Test Parameters

In order to test each probe, some tests have been defined. Each test has got different configuration parameters in order to test the probes in different situations.

The parameters that can be configured are packet size, injection rate, analysis load, number of CPU cores and test duration.

The analysis load is simulated implementing different loops that take different number of loops. In this paper, the results shown are made with 1000 processing loops and 25000 processing loops of analysis load.

Each test takes four minutes and it is made with the same traffic rate, packet size (54 bytes), analysis load and number of cores. A battery of tests is a group of tests with the same configuration parameters but the traffic rate that increases for each test from 50.000 packets per second up to 1.500.000 packets per second (1 Gbps with the fixed packet size).

There are tests for one, two, and four CPU cores. The machine used for running the probes in the tests has got two quad core processors. In the tests with two cores, there is one core running in each processor. On the other hand, in the tests with four cores, there are two cores running in each processor.

 ISBN: 978-1-61208-318-6

V. TEST RESULTS AND DISCUSSION

In order to test the probes, three test batteries have been done for each analysis load and for each probe and with different number of CPU cores. Each battery is composed of 21 tests of four minutes. Each test is done at a different rate.

Each graph in Fig. 6 shows the analysis throughput for the three probes with 1000 loops of analysis load and a fixed number of CPU cores. In Fig. 7, it can be seen, in each graph, the results for the three probes tested in this paper with 25000 loops of analysis and a fixed number of CPU cores. The graphs in both figures show the analysis throughput, that is, the throughput of the probe in packets per second. They have three series of data, one for each probe.

On the other hand, Fig. 8 and Fig. 9 show the capture throughput for the three probes with 1000 and 25000 loops of analysis load and a fixed number of CPU cores.

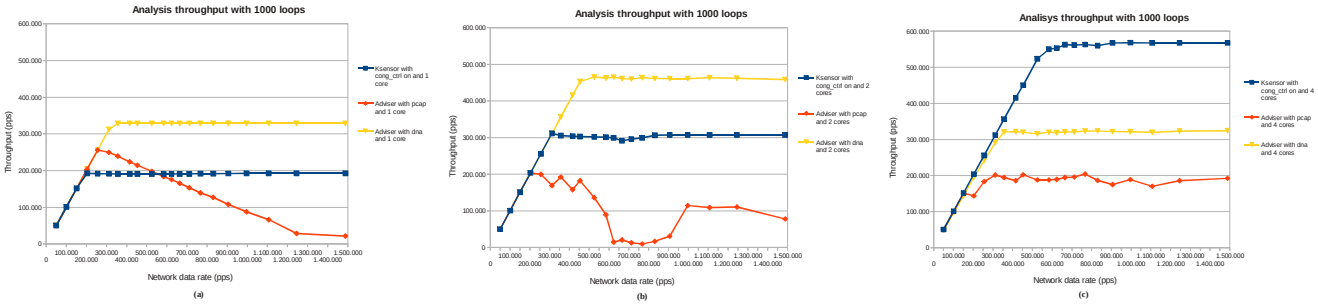


Figure 6. Comparison of analysis throughput with 1000 loops of analysis load. (a) With 1 CPU core. (b) With 2 CPU cores. (c) With 4 CPU cores.

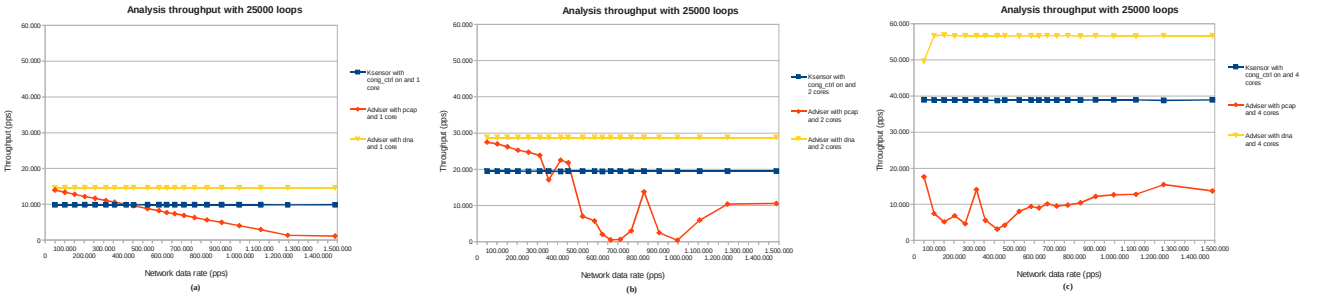


Figure 7. Comparison of analysis throughput with 25000 loops of analysis load. (a) With 1 CPU core. (b) With 2 CPU cores. (c) With 4 CPU cores.

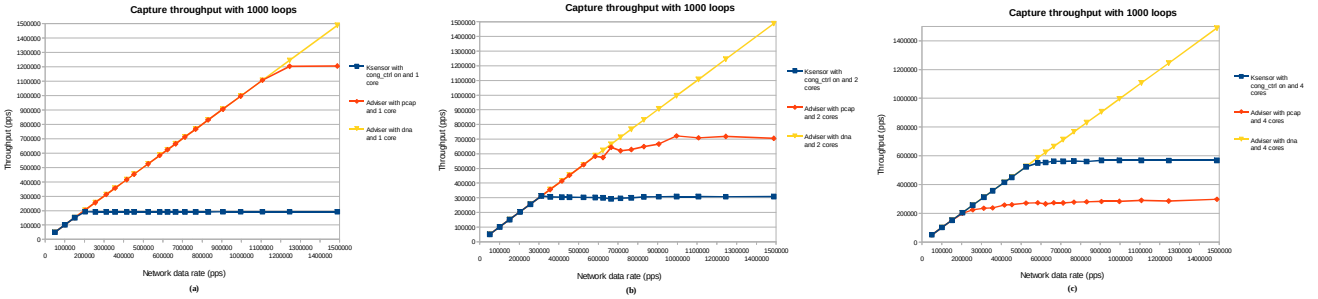


Figure 8. Comparison of capture throughput with 1000 loops of analysis load. (a) With 1 CPU core. (b) With 2 CPU cores. (c) With 4 CPU cores.

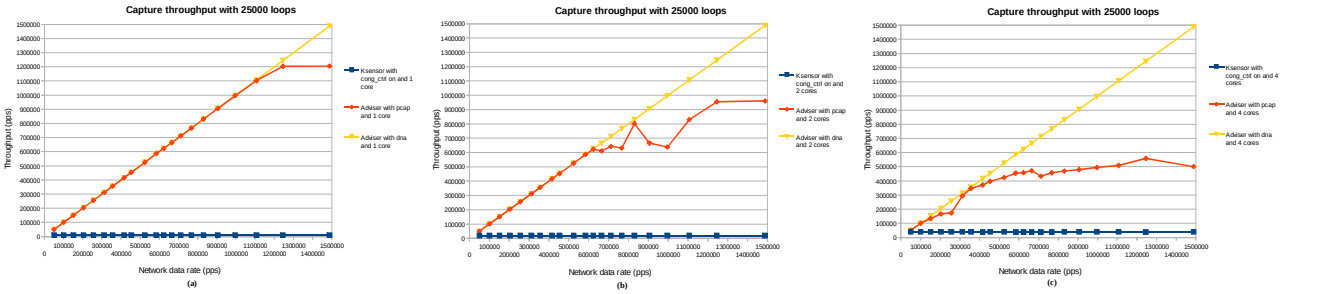


Figure 9. Comparison capture throughput with 25000 loops of analysis load. (a) With 1 CPU core. (b) With 2 CPU cores. (c) With 4 CPU cores.

 ISBN: 978-1-61208-318-6

It is remarkable that all these tests are done with 54 byte packets, the minimum sized ones that work in Ethernet networks. This means that, with a data rate of 1 Gbps the probes receive the maximum number of packets as possible. The system allocates its buffers taking into account the number of received packets and not the size of them.

This paper shows results of two prototypes with Adviser. One of them uses Libpcap as interface to capture packets, while the other prototype uses PF_RING_DNA. It is worth mentioning that the prototype with PF_RING DNA uses threads in order to implement the analysis task while the prototype with Libpcap uses processes. We also show results from Ksensor, the kernel-level probe presented before.

As we can see in Fig. 6-9, the prototype that has the worst performance is Adviser with Libpcap. With one CPU core it has a stable behavior. The analysis throughput is the lowest one although the capture throughput is nearly the same as with PF_RING DNA, the highest one. This happens because the capture processes have higher priority that the analysis ones. Besides, the packets are captured with all the infrastructure of the operating system. The packets are disassembled and treated as normal packets. Because of all this, the capture takes a lot of time.

When the system is capturing packets the analysis processes are slept and are not analyzing packets because there is only one CPU. Because the system takes more time capturing packets and the capture processes have more priority than the analysis ones, there are more captured packets than analyzed ones. This means that the system has to drop packets without analyzing them so the analysis throughput is lower than the capture one. There is a lot of CPU usage lost capturing packets that the system is not able to analyze. With more than one CPU core the behavior of Adviser with PCAP has the same problems that have been explained in the previous paragraph. Moreover, the design of this prototype has not resolved well the multiprocessor execution. It has two problems. The first one is that the design is done with processes. The system can execute only one process at a time so the system cannot execute more than one analysis task at the same time although the analysis processes have affinity with one CPU core. The second problem is that there is only one packet queue and the processes have to compete in order to take a packet from the queue. Because of all this, the behavior of the probe is not very stable and the performance is not good.

Obviously, the performance of the analysis with higher analysis load is lower. The system takes more time in analysis per packet so it analyzes fewer packets per second.

Regarding Ksensor, its congestion avoidance mechanism guarantees that all the packets that are captured are analyzed. Because of this, the capture throughput (see Fig. 8-9) and the analysis throughput (see Fig. 6-7) are the same.

On the other hand, we can observe in Fig. 8-9 that PF_RING DNA captures all the packets that are sent. In this case, the CPU does not execute anything because PF_RING DNA works with memory mapping.

If we compare the capture throughput of Adviser with PF_RING DNA and the capture throughput of Ksensor, we can see that PF_RING DNA has a better performance on capture terms. Moreover, the prototype with PF_RING DNA does not use CPU resources in order to capture packets so all the resources can be used to analyze them.

The comparison of the analysis throughput is not so easy. There is only one packet queue in both cases. Both prototypes have implemented threads for the analysis. So, with more than one core there are many consumers of the packet queue. There are many threads competing to access to the queue.

The higher the analysis load is the fewer accesses must be made to the packet queue. With high analysis loads the system analyzes fewer packets than with a lower analysis load. This means that, with a higher analysis load, the analysis threads make fewer accesses to the queue so there are fewer concurrency problems.

If we compare the analysis throughput we can see that, with 1000 loops of analysis, the performance of Ksensor with 2 cores is lower than the performance of the prototype with PF_RING DNA. On the other hand, with 4 cores, the performance of Ksensor is higher. With 25000 loops of analysis, the performance of the prototype with PF_RING DNA is higher in both cases, with 2 and with 4 CPU cores. One of the differences between 1000 and 25000 loops is that, with 1000, there are more accesses to the queue so the analysis threads have to wait more time in order to take a packet. Both prototypes work with as many analysis threads as CPU cores.

Ksensor has a better design for the multiple accesses to the packet queue with more than one thread at the same time but PF_RING DNA has a better performance in packet capture. With 1000 loops there are many accesses to the queue but the performance in analysis of Adviser with PF_RING DNA is higher with one and two CPU cores. But with four cores the performance of Ksensor is higher. With one and two cores the performance of the capture of Adviser with PF_RING DNA makes the analysis performance higher but with four cores the low performance in multiple access of the prototype Adviser makes the analysis performance be low. With 25000 loops there are fewer accesses to the queue so there are not as many problems as before with the multiple accesses to the queue.

Obviously, with more CPU cores the performance of the probes is higher.

VI. CONCLUSIONS

This work sets out to evaluate two software probes based on commodity hardware under different configurations. On the one hand, Adviser, a user-level framework, is evaluated with several current capturing systems (NAPI with LibPcap, PF_RING with DNA, PFQ) and several analysis loads (1000, 25000 processing loops). On the other hand, Ksensor, a kernel-level framework, uses NAPI in the capturing stage and it is tested for different analysis loads (1000 and 25000 processing loops too). It is worth mentioning that all the evaluations have been performed on the same hardware platform. It has got two quad core processors. When it is configured with one or two cores it uses one core per processor, but with more than two cores it has to use more than one core per processor. It is also remarkable the use of a

 ISBN: 978-1-61208-318-6

testing architecture which configures the tests, runs them and gathers the results automatically.

The results indicate that Adviser with NAPI-Pcap is not a good solution. Its behavior is not predictable and its performance is lower than the performance of the other probes. With low analysis load, the performance of Adviser with PF_RING-DNA with four cores is lower than the performance of Ksensor and, even, the performance of Adviser with PF_RING and DNA and two cores. With high analysis load, the performance of Adviser with PF_RING-DNA is higher than the performance of Ksensor.

All these results have their corresponding explanation. The numerous copies in the capturing process and the absence of a congestion control mechanism between the capturing and the analysis stage are the main reasons of the unstable behavior of Adviser with NAPI-Pcap. However, Adviser with PF_RING-DNA provides a higher performance due to the improvement that it offers in the capturing stage, although there could be concurrency problems. We are referring to the problems between the capturing and analysis instances when both of them try to access the same packet queue. Finally, Ksensor does not provide a capturing performance as good as PF_RING-DNA, but it incorporates elements of control to solve concurrency problems, as well as a congestion control mechanism. For this reason, under certain circumstances (for instance, the case of 4 CPU cores with 1000 loops analysis load), Ksensor can offer a better performance than PF_RING-DNA.

As a future work we plan to migrate the prototype Ksensor to a recent Linux version in order to take advantage of the improvements that this recent kernel offers in capturing performance. In this way, the adaptation of the probe to the Generic Receive Offload (GRO) and Receive Packet Steering (RPS) techniques, which are included in recent kernel versions, can bring benefits for the system performance. On the one hand, GRO implies to change the processing of the packets in the capturing stage, by grouping packets which belong to the same flow. On the other hand, RPS proposes to increase the number of packet queues, by having one packet queue for each processor and by creating a NAPI virtual interface for each processor. This will imply to reduce the concurrency problems between the capturing and the analysis instances.

As explained in Section III, PFQ has been integrated into Adviser. This has been validated by using a conventional NIC (in particular, the model Intel 82574L) and the results obtained have been similar to native PF_RING (without DNA). But PFQ needs a multiqueue NIC in order to obtain an optimal performance. As the test scenario described in Section IV does not have any NIC of this type, Adviser with PFQ has not been tested under the optimal conditions. For this reason, there is not any result of PFQ in the comparison of Section V. In the future, we plan to obtain a multiqueue NIC to test Adviser with PFQ properly.

Finally, we want to mention that, once the migration of Ksensor is completed, we also plan to make a new comparison among the new Ksensor, Adviser with PF_RING-DNA and Adviser with PFQ

ACKNOWLEDGMENT

We gratefully acknowledge support from the Basque Government funding the VMAT project within the SAIOTEK 2012 initiative in the scope of which this research work has been conducted.

REFERENCES

[1] F. Schneider, “Packet capturing with contemporary hardware in 10 Gigabit Ethernet environments," Proc. Passive and Active Measurement Conference (PAM 2007), Springer-Verlag Berlin Heidelberg, Apr. 2007, pp. 207-217.

[2] ntop project, http://www.ntop.org, 14.10.2013.

[3] A. Fiveg, “Ringmap capturing stack for high performance packet capturing”, http://wiki.freebsd.org/AlexandreFiveg, Sept. 2010.

[4] L. Rizzo, “Netmap: a novel framework for fast packet I/O,” Proc. 2012 USENIX Annual Technical Conference, USENIX Association, Jun. 2012, pp. 9-20.

[5] L. Deri, “Improving passive packet capture: beyond device polling,” Proc. 4th International System Administration and Network Engineering Conference (SANE), vol. 2004, Oct. 2004, pp. 85-93.

[6] L. Deri, “nCap: Wire-speed packet capture and transmission,” IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON), IEEE, May. 2005, pp. 47-55.

[7] F. Fusco and L. Deri, “High speed network traffic analysis with commodity multi-core systems,” Proc. Internet Measurement Conference (IMC 2010), ACM, Nov. 2010, pp. 218-224.

[8] A. Cardigliano, L. Deri, J. Gasparakis, and F. Fusco, “vPF_RING: Towards wire-speed network monitoring using virtual machines,” Proc. Internet Measurement Conference (IMC 2011), ACM, Nov. 2011, pp. 533-548.

[9] N. Bonelli, A. Di Pietro, S. Giordano, and G. Procissi, “On multi-Gigabit packet capturing with multi-core commodity hardware,” Proc. 13th Passive and Active Measurement Conference (PAM), Springer, Mar. 2012, pp. 64–73.

[10] T. Mrazek and J. Vykopal, “Packet capture benchmark on 1 GE”, CESNET technical report 22/2008, http://www.cesnet.cz, Dec. 2008.

[11] L. Braun, A. Didebulidze, A. Kammenhuber, and G. Carle, “Comparing and improving current packet capturing solutions based on commodity hardware,” Proc. Internet Measurement Conference (IMC 2010), ACM, Nov. 2010, pp. 206-217.

[12] A. Ferro, F. Liberal, A. Muñoz, I. Delgado, and A. Beaumont, “Software architecture based on multiprocessor platform to apply complex intrusion detection techniques”, Proc. 2005 IEEE International Carnahan Conference on Security Technology (CCST’05), IEEE, Oct. 2005, pp. 287-290.

[13] C. Benvenuti, Understanding Linux Network Internals, O'Reilly Media, 2005.

[14] LibPcap, http://www.tcpdump.org, 14.10.2013.

[15] PFQ Homepage, http:// netserv.iet.unipi.it/software/pfq, 14.10.2013.

[16] PFQ: Accelerated packet capture engine for multi-core architectures, http://pfq.github.com/PFQ, 14.10.2013.

[17] A. Munoz, A. Ferro, F. Liberal, and J. Lopez, “A kernel-level monitor over multiprocessor architectures for high-performance network analysis with commodity hardware,” Proc. SensorComm 2007, IEEE, Oct. 2007, pp. 457-462.

[18] A. Pineda, L. Zabala, and A. Ferro, “Network architecture to automatically test traffic monitoring systems,” Proc. Mosharaka Int. Conference on Communications and Signal Processing (MIC-CSP2012), Academy, Apr. 2012, pp. 18-23.

 ISBN: 978-1-61208-318-6

Efficient Performance Diagnosis in OpenFlow Networks Based on Active Measurements

Megumi Shibuya, Atsuo Tachibana and Teruyuki Hasegawa
KDDI R&D Laboratories, Inc.
Saitama, JAPAN
{shibuya, tachi, teru}@kddilabs.jp

***Abstract*—To detect performance degradation and identify causal links promptly in OpenFlow networks, this paper presents the design of a scheme to actively measure the performance of all physical links from a single measurement point based on the controllable feature of individual traffic flow provided in OpenFlow. In the scheme, the measurement paths from a measurement box (called a "beacon") are calculated to comprehensively cover all links and the probing packets are transmitted along elaborately defined routes in cooperation with the OpenFlow controller and multiple switches. Diagnosing the link performance of an entire OpenFlow network with a single beacon is expected to reduce operational costs, such as deployment and maintenance costs of beacons. In addition, reducing the number of flow-entries for active measurements on OpenFlow switches is considered to save the resources of OpenFlow switches. The effectiveness and feasibility of our solution are demonstrated through model emulations.**

Keywords- OpenFlow; Active measurement; Network Diagnosis

I. INTRODUCTION

Software Defined Networking (SDN) architectures that offer flexible traffic control based on programmable network functionalities have been studied. OpenFlow [1] is a major example of the SDN architectures used to control the traffic flows of multiple switches from a centralized controller. In order to provide high quality services to customers over OpenFlow networks, it is important for network operators to manage the networks in a reliable and efficient manner, and performance characteristics and states of network links should be measurable and tractable.

Unfortunately, standard management methods, such as periodically monitoring all network-internal devices using the Simple Network Management Protocol (SNMP), are not always effective for promptly detecting network performance degradation. For example, it is not easy to detect performance degradation of packet delay by analyzing the set of status and event information monitored at individual devices inside the network. In addition, the failure of SNMP-based detection is potentially caused by bugs in router/switch software and network misconfiguration [2]. Accordingly, measuring link performance by actively probing the network is important as an alternative and/or complementary method to detect performance degradation.

With the aim of identifying the root cause of performance degradation in conventional networks, a variety of network diagnostic tools that estimate link-by-link performance characteristics, such as distribution of delays, loss rate and bandwidth, have been developed by exploiting the ICMP responses generated by routers to specially crafted probing packets. However, the probing packets are usually transmitted along pre-defined paths (the shortest paths in most cases), and hence, to comprehensively monitor the entire network, multiple measurement boxes (hereafter, called "beacons") need to be distributed [3][4]. Therefore, it is not always cost efficient from the viewpoint of network operators [5].

In this paper, we propose a diagnosis scheme that utilizes the controllable feature of individual traffic flow provided in OpenFlow networks and conducts an existing active measurement tool to comprehensively monitor the link performance in the OpenFlow network by using a single beacon in cooperation with the OpenFlow controller and multiple switches. In the proposed scheme, each OpenFlow switch maintains some exclusive flow-entries dedicated to active measurements in addition to the user's data traffic flows so that measurement paths (flows) cover all links in the network. According to the flow-entries at each switch, the probing packets are forwarded to the designated physical links and eventually return to the beacon. The beacon estimates link-by-link performance characteristics by analyzing the difference between probing packets returned from different switches based on an existing (*traceroute*-like) measurement technique.

The contributions of this work are two-fold. First, we present the design of a diagnosis scheme to actively measure the performance of all physical links from a single measurement point in the OpenFlow network. Note that we currently use a traceroute-like measurement technique, but we believe we could use other active measurement tools without difficulty. Second, we propose a method to reduce the number of exclusive flow-entries for active measurements on OpenFlow switches by aggregating multiple entries into a single entry by applying common packet header options to the probing packets. Because these excursive flow-entries themselves consume the limited resources of flow table memory of the OpenFlow switch, the overhead flow-entries should be minimized. The feasibility and effectiveness of the proposed method are verified through emulations with Trema [6].

This paper is organized as follows. Section II shows related work on the existing network measurement techniques. Section III explains the proposed measurement scheme on how to define the measurement paths using only one beacon. In Section IV, we evaluate and discuss the effectiveness and the feasibility of the proposed scheme

 ISBN: 978-1-61208-318-6

through the emulations. Finally, we conclude the work in Section V.

II. RELATED WORK

A variety of network diagnostic tools that estimate link-by-link network characteristics, such as loss rate, distribution of delays, and bandwidth, along a network path have been extensively developed by sending probing packets over ICMP or specially crafted TTL-limited probing packets, and exploiting the responses generated by routers along the measurement paths (e.g., traceroute command, *Pathneck* [3], *cing* [7], and *BFind* [8]). Network tomographic approaches have also been proposed to infer the performance characteristics of the network interior accurately and/or in identifying degraded links solely based on the correlation among multiple end-to-end path measurements (e.g., a survey [9]). In these studies aiming at conventional networks, each probing packet is assumed to pass though pre-defined paths (the shortest paths among sender and receiver beacons in most cases) and hence, multiple beacons are required to comprehensively monitor the entire network. To effectively monitor large-scale networks, good placement of beacons (a minimal set of beacons) and measurement paths among them is considered with some operational constraints (e.g., [10][11]). For example, in reference [10], selecting optimal sets of active measurement paths to cover all the links is considered with the measurement load constraints and installation costs. However, to the best of our knowledge, few researches have been conducted on the design and development of efficient solutions to promptly detect performance degradation and identify causal links in OpenFlow networks.

III. PROPOSED METHOD

A. Calculating Measurement Paths

Unlike conventional Layer 2 and 3 switches, the OpenFlow switches can define *flows* based on the combination of Layer 1, 2, 3, and 4 information such as the ingress port of switches, the MAC address, IP address and port number involved in the incoming frame/packet header fields, and *actions* on how the switches handle the received packet/frame (flow), such as packet forwarding to the designated links (switch ports) and packet header rewriting.

By utilizing the controllable feature of each traffic flow provided in OpenFlow networks, in this paper, we propose a scheme that calculates the measurement paths to comprehensively cover all links from a single beacon and set the calculated paths along which the probing packets are forwarded to multiple switches in cooperation with the OpenFlow controller. In the proposed scheme, the packet header is rewritten at the designated switches to enable the probing packets to be returned to the beacon for the estimation of the performance of individual links. Consequently, the flow-entries for performance measurements at OpenFlow switches define three types of actions, packet-forwarding in the direction from the beacon to other switches (hereinafter called the "forward direction"), sending back received packets to the ingress links and forwarding packets in the direction from other switches to the beacon (hereinafter called "return direction"). The measurement paths are calculated as follows. Here, we suppose the detailed physical topology of the network is given.

1. Suppose a single beacon is connected to a switch in an OpenFlow network that contains N links (switch ports) $\{l_1, l_2, ..., l_N\}$. The shortest paths $\{r_1, r_2, \ldots, r_N\}$ from the beacon to all links are calculated by using an existing graph search algorithm (e.g., Dijkstra algorithm) based on a certain routing criterion (e.g., the number of hops from the beacon). Here, note that the shortest paths are not calculated based on switch-level topologies but link-level (i.e., switch port-level) topologies. In the case where there are multiple shortest paths to a link due to Equal-Cost Multi-Path (ECMP), one of them is selected. The calculated paths are adopted as the measurement paths in the forward direction.
2. In order to cover uncovered links, additional measurement paths are calculated. Suppose an uncovered link l_a between two switches each of which is connected by links l_b and l_c ($b \neq c$) in the forward direction paths, respectively, we compose measurement path r'_a by combining either of the two shortest paths r_b or r_c, and the uncovered link l_a. For example in Fig. 1, r'_5 is composed of r_3 (l_1-l_2-l_3) and l_5 as illustrated by the red dotted arrow in Fig. 1.
3. Against each path in the forward direction, the measurement path in the return direction is calculated as the path includes the same links and switches in reverse order. The paths in both the forward direction and return direction are combined together as a (round-trip) measurement path. As a result, the set of measurement paths that cover all links in the network is calculated.

B. Setting Measurement Paths in OpenFlow networks

As explained in Section III-A, in OpenFlow networks, traffic flows can be flexibly defined based on the combination of multiple packet headers, and hence there are many ways to define the flows for the performance measurements. As a simple example, we can define the flows by using two-tuple fields in the packet header, destination port number and destination address. An example setting of flow definition and actions on switches are as follows.

1. The beacon sends probing packets that have different destination port numbers toward every target link (i.e., switch ports that are connected to target links) that will send back the probing packets.
2. When an OpenFlow switch receives a probing packet, the switch identifies which flow-entry is matched for the probing packet based on the destination port number, and applies pre-defined

 ISBN: 978-1-61208-318-6

regarding actions on how the switches handle the received packet. If the probing packet is identified as the traffic flow for performance measurements in the forward direction, the probing packet is forwarded to the designated link toward the next switch according to the pre-defined action in the flow-entry table.

3. If the probing packet is identified as flows to be sent back to the beacon based on the destination port number, the switch forwards the probing packet to the ingress link. In order to make other switches distinguish the directions of the probing packets (i.e., forward direction and return direction), the switch changes the destination address of the probing packets to a certain unique address (e.g., the IP address of the beacon). Note that packet rewriting and packet-forwarding are defined by a single flow-entry at OpenFlow (version 1.3) switches.
4. As a result, the probing packets in the return direction are identified by the destination address of the probing packets at any switch and the packet-forwarding action in the return direction is conducted at multiple switches along the measurement paths.
5. As a result, the beacon observes all the returning probing packets from all individual links and estimates link-by-link performance characteristics by analyzing the difference between probing packets as the traceroute command does. Note that we currently use a simple estimation technique of link-by-link performance, but we believe that other sophisticated techniques are widely applicable to our scheme.

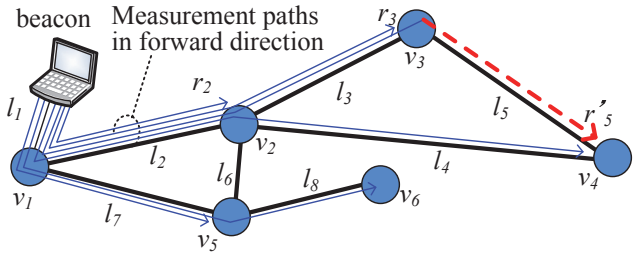


Figure 1. The calculation of the measurement paths.

C. Aggregation of Measurement Flows

In OpenFlow networks, setting a large number of flow-entries at a switch may cause problems where flow-entries dedicated to user traffic flows cannot be added due to the limitation of the maximum number of flow table entries [12]. In addition, the performance of packet processing at a switch may be degraded [13][14][15]. Therefore, it is important to minimize the number of flow-entries for performance measurements in each switch to the extent possible.

In the proposed scheme, flow-entries for performance measurements kept at switches are classified into three types, packet-forwarding in the forward direction, sending-back the received packets toward the beacon, and packet-forwarding in the return direction. However, in this section, we focus on reducing the number of flow-entries of measurement paths in the forward direction. This is because (i) measurement paths in the return direction can be easily aggregated by setting a common header option in all returning probing packets. For example in Section III-B, each switch can define flow-entries for multiple paths in the return direction by a single flow-entry based on the same identifier (e.g., the destination address) of a flow and the same packet-forwarding action. (ii) sending-back action can be merged with packet-forwarding action in the current implementation of OpenFlow switches.

Aiming at reducing the number of flow-entries at switches in the network, we propose a solution that utilizes unused header fields of probing packets. By setting the common header option to multiple measurement flows that define the same action, we aggregate multiple flow-entries into a single flow-entry. For example in Fig. 2, if we set the same source port number to all the probing packets that pass through link l_3 in the forward direction, i.e., r_3 and r'_5, the two flow-entries for these flows can be merged by a single flow-entry at switch v_2, i.e., switch v_2 can identify the flow by using the source port number and apply the same action to forward the packet to link l_3. Here, it should be noted that the same aggregation of the flows is applicable to other switches that a same set of aggregated flows traverse. Namely, the number of flow-entries at switch v_1 is also reduced in Fig. 2.

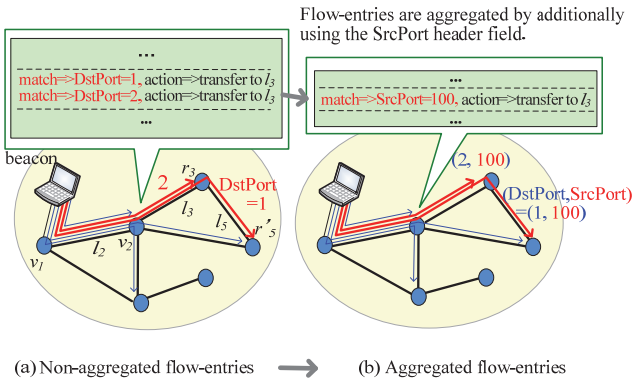


Figure 2. Flow aggregation of the flow-entries.

The problem of finding optimal sets of flows to reduce the number of flow-entries over the network is formulated as follows. Suppose the monitored OpenFlow network consists of M switches $\{v_1, v_2, \ldots, v_M\}$ and N links (switch ports) $\{l_1, l_2, \ldots, l_N\}$, and the number of the flow-entries at v_i is expressed by F_i (i=1, 2, …, M). As the number of flow-entries at a switch may affect the performance of the packet processing and the number of available flow-entries for user data traffic, in this paper, we focus on minimizing the

 ISBN: 978-1-61208-318-6

maximum number of flow-entries at all switches in the network as expressed in (1).

$$\text{Minimize} \qquad \max\{F_1, F_2, \ldots, F_M\} \tag{1}$$

Here, F_i (i=1, 2, …, M) is calculated by the sum of the number of flows that pass in v_i as formulated by (2). In (2), $l_j \in v_i$ means that link l_j is connected to switch v_i in the forward direction.

$$\text{s.t.} \qquad \forall i, \quad F_i = \sum_{l_j \in v_i} f_j \tag{2}$$

In addition, suppose the number of flows that pass through link l_j in the forward direction is expressed by f_j, the relationship among f_j (j=1, 2, …, N) can be formulated as linear equation (3).

$$\text{s.t.} \qquad [f_1, f_2, \ldots, f_L]^T = \boldsymbol{R} \cdot [f_1, f_2, \ldots, f_L]^T + [1, 1, \ldots, 1]^T \tag{3}$$

where **R** is a matrix (referred to as *routing matrix*) that associates the number of input flows and the number of output flows across the switches by reflecting the topology of the measurement paths. Routing matrix **R** = (a_{jk}) encodes whether link l_j is connected with l_k across a switch, i.e., a_{ij} takes a value of 1 if link l_i is connected with link l_j across a switch, and 0 otherwise. Note that the second clause on the right side in (3) represents the number of flows that will be sent back to the beacon.

Furthermore, in (4), we introduce the binary functions b_j (j=1, 2, ..., N), that indicate if all flows that traverse each link l_j are aggregated. Note that because the aggregation of flows is applied to reduce the flow-entries at switches, candidate sets of the aggregated flows pass through the common links. The total number of flows can be expressed by using b_j as shown in (5). In (5), f_j becomes 1 if the value of b_j is 1 and f_j otherwise.

$$\text{s.t.} \qquad b_j = \begin{cases} 1 & \textit{all flows that traverse link } l_j \\ & \textit{are aggregated} \\ 0 & \textit{otherwise} \end{cases} \tag{4}$$

$$f_j = (1 - b_j) f_j + b_j \tag{5}$$

Next, other constraints are considered. In our aggregation scheme, since we assume that multiple flow-entries are aggregated at switches based on the same header values of incoming probing packets, the values are not changed by any switches while the probing packets traverse the network in the forward direction. This is because changing the header values for the flow aggregation requires setting the exclusive flow-entries whose number is the same as the number of individual flows to be changed, and thus it contradicts our goal of reducing the number of flow-entries on measurement paths. In other words, when we use a single packet header field for the aggregation, measurement paths in the forward direction can be aggregated at most once. The constraints are formulated by (6) with the number of available header fields for the aggregation H.

$$\text{s. t.} \qquad \forall k, \quad \sum_{r_j \subseteq r_k} b_j \leq H \tag{6}$$

where $r_j \subseteq r_k$ means that two paths r_j and r_k are in the inclusion relationship.

The above equations, (1)-(6) are expressed as an integer programing problem. The optimal solutions (the minimized number of $\max\{F_1, F_2, \ldots, F_N\}$ and b_j (j=1, 2, …, L)) that achieve the minimization are calculated by a general programing solver (e.g., IBM ILOG CPLEX [16] and Gnu Linear Programming Kit (GLPK) [17]). After calculating the solution, all measurement flows traversing the relevant links are aggregated at each link l_j of b_j=1. A unique value is set to the header field in the aggregated probing packets. The new aggregated flow-entries are set in substitution for the existing flow-entries at the relevant switch in cooperation with the OpenFlow controller.

IV. Evaluation and Discussion

A. Experimental Setup

In order to evaluate the effectiveness and feasibility of our proposal, we set up experimental virtual networks by using *Trema* (v0.3.19) [6] of the OpenFlow emulator. Since the current Trema does not have the functions of accurately capturing packets and emulating the performance degradation, such as packet loss and large packet delay, we set up OpenFlow networks over two physical Linux servers (Servers 1 and 2) and two physical links between them as shown in Fig. 3. The experimental network is configured as follows.

- One host as a beacon was set up on server 2 by using Kernel-based Virtual Machine (KVM) [18]. All probing packets exchanged by the host are captured at the physical network interface between Servers 1 and 2 by using *tcpdump* command. We implemented the functions to collect the statistical data of each measurement flow at all links such as the number of received packets, average packet delay and loss rate on individual links, by analyzing recorded logs in Trema and packet captured data by tcpdump.
- To emulate the performance degradation on OpenFlow networks, we distributed OpenFlow switches over two physical servers and caused 5% packet loss or 20ms packet delay at the two physical links overlaid by the experimental OpenFlow network by using a network emulator [19].
- In order to measure packet delay in the millisecond order, we modified Trema to output the minimum logs. As a result, we confirmed that the packet transfer delay at each switch was 0.4ms on average. The specification of the two servers is as follows: CPU: Intel Xeon E5-2420 @ 1.90 GHz, memory: 18GByte, OS: Ubuntu v12.04.2).

 ISBN: 978-1-61208-318-6

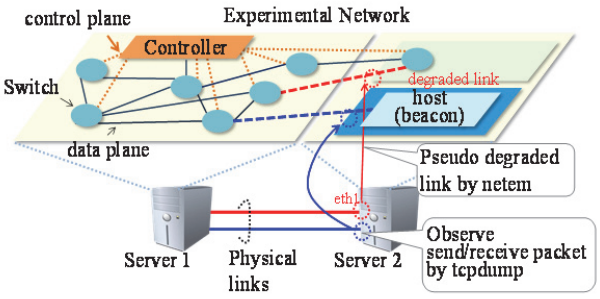


Figure 3. Experimental setup.

We used three mesh topologies and three tree topologies based on six random topologies of the Waxman model and BA model by BRITE [20] as shown in Table I. Here, we set the maximum number of switches to 300 due to the limitation of the current version of Trema. On each topology, we roughly classified all switches into three types (root-, center- and edge-positioned switches) based on the degree of each vertex (i.e., the number of connected links) and connected a beacon (host) to a switch of each type and repeatedly conducted emulations, i.e., we conducted three emulations on each topology.

TABLE I. EXPERIMENTAL NETWORK TOPOLOGY MODELS

No.	#Switches	#Links	Topology
1	10	18	Waxman, Mesh
2	20	21	BA, Tree
3	100	198	BA, Mesh
4	150	601	Waxman, Mesh
5	200	201	Waxman, Tree
6	300	300	BA, Tree

B. Results

1) Network measurement coverage

First, in order to verify the effectiveness of our proposed method, the host continuously sent the probing packets (64 Byte UDP, 100ms constant interval) to estimate the performance of all links in the network. At each link, the average packet delay and loss rate are estimated every ten seconds. The results are summarized as follows.

- First, we confirmed that all links were comprehensively covered by measurement paths and all probing packets were returned to the beacon based on the collected statistical data.
- In the case where we caused 5% packet loss and 20ms packet delay on a targeted link by using a net emulator, the beacon successfully detected the performance degradation on the link. The estimation error of loss rate and packet delay was approximately 10% and 15%, respectively. Note that the estimation accuracy depends on the measurement techniques and evaluation with a more sophisticated measurement method is a future study consideration.
- We confirmed that the results of six network topologies and three types of positions of the beacon's connection (totally 18 emulations) were almost the same.

2) Reduction of the number of flow-entries

To evaluate the effectiveness of the flow aggregation explained in Section III-C, we investigated the maximum number of flow-entries at a switch in the network in three cases; in the case where no flow-entries are aggregated (Case 1), in the case where flow-entries of measurement paths in the forward direction are aggregated (Case 2), and in the case where flow entries of measurement paths in both the forward direction and return direction are aggregated (Case 3). Figure 4 shows the ratio of the flow-entries between Cases 1 and 2, and between Cases 1 and 3, on different topologies and different positions of a switch connected by the beacon (root-, center- and edge-positioned switches). Through all emulations, the maximum number of flow-entries was successfully reduced compared with Case 1. In particular, in the case where the beacon is connected to a center-positioned switch on topology 6, the maximum number of flow-entries was reduced from 1201 to 276 (the reduction ratio was 0.78).

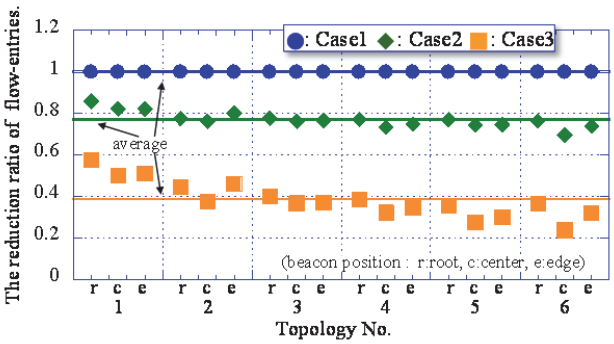


Figure 4. The ratio of flow-entries.

3) Computation time

We evaluated the average computation time of the calculation of measurement paths through emulation experiments. To calculate a set of measurement paths to cover all links on topology 6 by using the Dijkstra algorithm, our method took about 0.45s. In contrast, the computation time to calculate the optimal solution of the flow aggregation was approximately 0.1s. As a result, the computation time was sufficiently short for practical use on a network with 300 switches.

C. Elimination of Redundant Probing Packets

Sending probing packets itself is unsolicited and may cause network congestion especially on a large scale. Therefore, probing packets should be sufficiently light. Especially in the case where we utilize measurement tools that send not only probing packets but also load packets to find a bottleneck link and/or estimate link-by-link available bandwidth, it is important to minimize the number of probing packets. For example, BFind [8] detects bottleneck links by injecting a considerable volume of load traffic into

 ISBN: 978-1-61208-318-6

the network path, and at the same time, conducts traceroute to monitor the RTT changes to all links on the path. Pathneck [3] detects the location of the bottleneck by sending probing packet trains including load packets, and the bottleneck link is inferred by measuring the per-hop train length. Note that these tools usually estimate the link-by-link performance based on the per-path measurement.

To reduce the network load of probing packets, the beacon should remove redundant packets. For example in Fig. 1, switch v_1 is shared by many measurement paths toward the same switch r_2, r_3 and r'_5. In this case, if we can utilize the same load packets toward l_5 for measuring the performance of l_2 and l_3, the load packets to measure the performance of l_2 and l_3 can be eliminated. Therefore, clustering the load packets to be sent based on the route, and eliminating the duplicated load packets is a promising approach to reduce the number of probing packets.

To evaluate this approach, we conducted preliminary experiments by using the six topologies shown in Table I. We investigated the number of probing packets in two cases. In one case (Case 1), all performance measurements are separately conducted per path but redundant packets are not eliminated. In the other case (Case 2), redundant packets are eliminated after clustering measurement paths based on their inclusion relationship. Table II presents the results. As can be seen, in each topology, the number of probing packets was reduced by almost half, and we confirmed that the elimination method based on paths' clustering has considerable potential to reduce the traffic load of probing packets of the per-path measurements. Note that the reason why the reduction ratio is near 0.5 is that most of the measurements on a measurement path can be used for the calculation of the performance of two adjacent links based on the difference with the other measurements on the same path.

TABLE II. NUMBER OF PROBING PACKETS

Topology No.	#Probing packets		Reduction ratio (Case2 / Case1)
	Case1	Case 2	
1	34	18	0.53
2	40	21	0.53
3	394	198	0.50
4	1198	601	0.50
5	400	201	0.50
6	598	300	0.50

V. CONCLUSION

In this paper, we proposed a diagnosis scheme to actively measure the performance of all physical links from one measurement point in OpenFlow networks. In addition, we tackled to reducing the number of flow-entries at OpenFlow switches by aggregating multiple flow-entries into a single entry by applying common header options to probing packets, in order to save the resources of OpenFlow switches. The optimization solution is calculated by solving the integer programing problem. Through emulation with the Trema emulator, we confirmed the proposed method has considerable potential to efficiently measure the performance of all links in the OpenFlow networks. Furthermore, we consider other measurement techniques and/or operational constraints, such as the measurement load of networks in OpenFlow networks, and conducting more large-scale experiments in future work.

REFERENCES

[1] Open Networking Foundation, http://www.openflow.org/.

[2] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman, "IP forwarding anomalies and improving their detection using multiple data sources," Proc. ACM SIGCOMM, September 2004.

[3] N. Hu, L. Li, Z. Mao, P. Steenkiste, and J. Wang, "Locating Internet bottlenecks: Algorithms, measurements, and implications," Proc. ACM SIGCOMM, September 2004.

[4] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet path diagnosis," Proc. ACM SOSP, October 2003.

[5] A. Tachibana, S. Ano, and M. Tsuru, "A large-scale network diagnosis system based on user-cooperative active measurements," Int. J. Space-Based and Situated Computing, Vol. 3, No. 2, pp.69-82, 2013.

[6] Trema, http://trema.github.io/trema/.

[7] K. Anagnostakis, M. Greenwald, and R. Ryger, "Cing: Measuring network internal delays using only existing infrastructure," Proc. IEEE INFOCOM conference, 2003.

[8] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area Internet bottlenecks," Proc. SIGMETRICS '03, June 2003.

[9] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," Statistical Science, Vol. 19, No. 3 pp.499-517, 2004.

[10] Y. Zhao, Z. Zhu, Y. Chen, D. Pei, and J. Wang, "Towards efficient large-scale VPN monitoring and diagnosis under operational constraints," Proc. IEEE INFOCOM 2009, pp.531-539, April 2009.

[11] D. Ghita, H. Nguyen, M. Kurant, K. Argyraki, and P. Thiran, "Netscope: Practical network loss tomography," Proc. IEEE INFOCOM, March 2010.

[12] N. Matsumoto, M. Hayashi, and I. Morita, "LightFlow: Leveraging combination of hash and wildcard tables for high performance flow switching in large number of flow entries," USENIX 2013.

[13] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks," Proc. PAM 2010, pp.201-210, April 2010.

[14] N. Varris, and J. Manner, "Performance of a software switch," Proc. HPSR 2011, July 2011.

[15] A. Bianco, R. Birke, L. Giraudo, and M. Palacin, "OpenFlow switching: Data plane performance," Proc. ICC, pp.1-5, IEEE 2010.

[16] CPLEX, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[17] GLPK, http://www.gnu.org/software/glpk/.

[18] KVM, http://www.linux-kvm.org/page/Main_Page.

[19] S. Hemminger, "Network Emulation with NetEm," http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.

[20] A. Medina, A. Lakhina, I. Matta, and J. Byers. "BRITE: Boston University Representative Internet Topology Generator," http://cs-pub.bu.edu/brite/index.html.

 ISBN: 978-1-61208-318-6

Evaluating the Trade-off Between DVFS Energy-savings and Virtual Networks Performance

Fábio Diniz Rossi, Marcelo da Silva Conterato,
Tiago Ferreto, César A. F. De Rose
Faculty of Informatics
Pontiphical Catholic University of Rio Grande do Sul (PUCRS)
Porto Alegre/RS – Brazil
{fabio.diniz, marcelo.conterato}@acad.pucrs.br, {tiago.ferreto, cesar.derose}@pucrs.br

***Resumo*—Data centers usually employ virtualization techniques coupled with other techniques, such as Dynamic Voltage and Frequency Scaling (DVFS), in order to reduce overall energy consumption. However, changes in processor frequency may impact the network performance, specially in metrics such as throughput and jitter. This paper evaluates the trade-off between changes in processor frequency and network performance. Our results show that there is an opportunity to save energy by up to 15%, through the processor frequency reduction. However, this reduction in frequency may increase the response time of applications by up to 70%, directly influencing the quality of experience (QoE).**

***Keywords*—Benchmarking; DVFS; throughput; virtualization.**

I. INTRODUCTION

Cloud computing aims at providing scalable and on-demand IT resources (e.g., processing, storage, database) through the Internet. These resources can be accessed from anywhere, anytime, using any sort of computing device, such as desktops, tablets or smartphones. The market movement towards cloud computing and IT services outsourcing favors the business of data centers, but the segment still faces major challenges, particularly regarding capital expenses and power consumption costs.

According to a report from Stanford University [1], power consumption in data centers has increased significantly in the last years. Between 2005 and 2010, energy consumption increased by 56% around the world (36% only in the United States). Beyond economics, energy consumption affects other issues, such as cooling and emission of harmful gases.

To enable energy-savings, new proposals have been presented from green data center designs, using natural air cooling, to the use of special technologies that optimize resources utilization. Virtualization [2], [3], [4] is one of these technologies, which serves as the core infrastructure of current cloud computing environments, and due to its features such as virtual machines migration and server consolidation, enables reduction in energy consumption. In addition, there are also technologies that allow energy-savings in data center servers, putting servers in standby or altering processing performance to adequate workloads demand, and consequently decreasing energy consumption.

In particular, Dynamic Frequency and Voltage Scaling (DVFS) [5], [6] is a technique frequently used to save energy on servers. DVFS is specially interesting in data centers that employ virtualization, where each server hosts a different group of virtual machines with diverse aggregate resources demands. However, several studies, such as [7], [8], show that changes in processor frequency can directly impact on the performance of network-dependent applications. This can be a decisive factor for the utilization of DVFS in data centers that support cloud services, since when the processor frequency is reduced, the processing capacity of the node is compromised, affectting all other components, including the network.

Clearly, there is a trade-off between using DVFS to save energy and network performance, which can directly impact on applications' Quality of Service (QoS) and Service Level Agreement (SLA). In addition, an important factor that may be impacted by this trade-off is the Quality of Experience (QoE). This parameter allows to measure the overall application performance from the users' point of view, showing a personal satisfaction perspective of the service offered by the service provider.

This paper aims to verify the impact of DVFS policies on network intensive applications performance running on a virtualized infrastructure (Citrix XenServer). The experiments were performed using three different DVFS policies, covering all possible configurations of processor frequencies allowed. The experiments were performed using a synthetic benchmark simulating a web application, in which an external client performs multiple requests through the network.

This paper is organized as follows: Section II introduces networking in virtualized environments and the DVFS technique; Section III presents related work; Section IV describes the testbed, the evaluation method and results obtained; finally, Section V concludes the paper and addresses future works.

II. BACKGROUND

This section introduces the concepts of virtual networks and IO management in virtualized environments, in particular the Citrix XenServer, which is the platform used in our experiments. Finally, we show how the changes of processor frequency are performed by DVFS.

 ISBN: 978-1-61208-318-6

A. Virtual Networks and IO Management

Virtualization is a technique that allows the sharing of computer resources between virtual machines, each one hosting a complete operating system. Virtual machines management is performed by the Virtual Machine Monitor (a.k.a hypervisor). Each virtual machine has one, or more, virtual network interfaces, used to communicate with adjacent virtual machines (located in the same server) or machines located elsewhere.

In this paper, we use the Citrix XenServer environment in our experiments. Citrix Xen Server is a free virtualization platform, suited to build cloud infrastructures. It uses the Xen hypervisor as the core component of its architecture to provide a stable and elastic abstraction of the underlying infrastructure. In this section, we focus on two important points that may influence the network overhead: virtual network architecture and IO management.

Xen's architecture [9] is composed of a special virtual machine, called domain 0 (dom0), which is responsible for managing all other virtual machines, called domain U (domU). Dom0 has also privileged access to IO devices. The other virtual machines (domU) host regular operating systems and each one has a virtual driver which communicates with Dom0 in order to access physical IO devices.

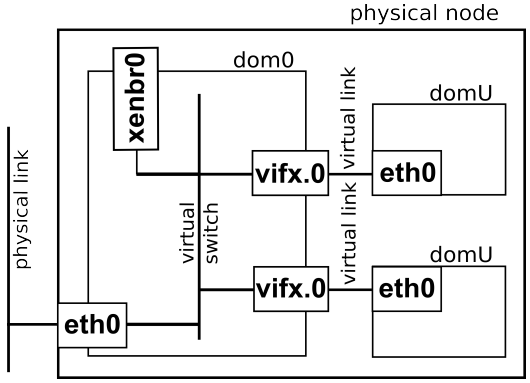


Figura 1. Virtual Network Architecture

Figure 1 shows how the virtual network is configured in Xen. Each virtual machine (domU) provides a complete hardware infrastructure, even if some devices do not exist physically or are shared by multiple virtual machines. An example of these devices is the virtual network adapter. The hypervisor may create one or more vifs (virtual interfaces) for each virtual machine, connected to a virtual link. Vifs are treated as regular NICs by the virtual machine, but in fact they only represent the interface for the physical NIC. These virtual and real networking components are connected with the use of a virtual switch. The hypervisor allows the construction of dynamic virtual network switches to enable communication between the virtual machines. Finally, the hypervisor also enables communication with the physical network infrastructure connecting the physical NICs of the server to the logical infrastructure of the hypervisor, enabling efficient communication between virtual machines, as well as with the external network.

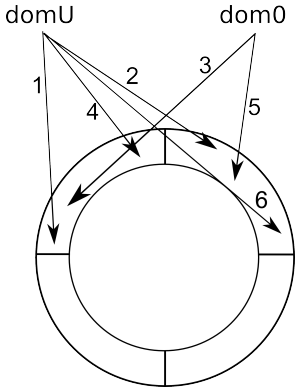


Figura 2. Ring Buffer Operations

IO is controlled by Xen through ring buffers. The data exchanged between dom0 and domUs in memory are controlled by a ring structure based on the producer-consumer model. This allows a model of locking in which there are two types of operations: request and response. Figure 2 shows how the communication occurs between dom0 and domU: (1) domU writes in the buffer a first request; (2) domU writes in the buffer a second request; (3) dom0 writes on buffer the response to the first request; (4) domU reads the dom0 answer about the first request and frees the buffer; (5) dom0 writes on buffer the response to the second request; (6) domU reads the dom0 answer about the second request and frees the buffer.

B. DVFS

Idle nodes still consume energy. Dynamic Voltage and Frequency Scaling (DVFS) is a technique that provides automatic adjustment of processor frequency with the intention to save energy. To make it happen, the processor must be able to operate in a range of frequencies, and these are adjusted according to processor utilization. Reducing the operating frequency reduces the processor performance and the energy consumption. Furthermore, reducing the voltage decreases the leakage current from the CPU's transistors, making the processor more energy-efficient resulting in further gains. Adjusting these parameters may result in a significant reduction in energy consumption per second.

Changing processor frequency decreases the number of instructions that can be executed per second, reducing overall server performance. Therefore, DVFS are usually not suitable for processes that are CPU-intensive. DVFS can be set by various operating policies such as:

- Performance: the frequency of the processor is always fixed at the highest, even if the processor is underutilized.
- Ondemand: the frequency of the processor as adjusted according to the workload behavior, within the range of frequencies allowed.
- Powersave: the frequency of the processor is always fixed at the smallest allowable frequency.

 ISBN: 978-1-61208-318-6

- Conservative: it has the same characteristics of the *On-demand* policy, but frequency changes are controlled, scaling gracefully between minimum and maximum according to processor utilization.
- Userspace: this allows setting a policy for each process in user space.

In order to evaluate the trade-off between DVFS operating policies and network throughput, the evaluations were performed with the three main policies: *performance*, *ondemand*, *powersave*.

III. Related Work

DVFS is a technology widely discussed in recent studies. It enables the reduction of processor frequency in order to save power when the processor utilization rate is not high, or even at times when the utilization rate changes over time.

In Takouna et al. [10] is shown that there are energy savings in virtualized clusters when used together DVFS and virtual machines consolidation. As an attempt to reduce the trade-off between energy consumption and average acceptance of jobs, a power consumption model was developed based on the number of cores, average processor frequencies and memory usage. The results show better energy savings, both in comparison with only DVFS, and DVFS with virtual machines consolidation.

Lago et al. [11] presents a strategy for resource allocation of virtual machines in a virtual cluster environment. The main focus of the work is the placement of the virtual machines in order to provide better cooling of the cluster, while DVFS is used as an alternative to decrease cooling requirements of each node individually.

The work of Belograzov et al. [12] proposes a linear interpolation model for predicting energy-savings of DVFS in cloud computing environments. The authors developed a model on the CloudSim simulator [13] in order to show the energy savings of correctly placing virtual machines and the impact of the ondemand DVFS policy. In Beloglazov and Buyya [14], the authors complement the previous work proposing a heuristic for placement of virtual machines with the intention to save energy, while meeting QoS requirements.

Kanga [15] proposed a change in the resource scheduler of Xen in order to optimize energy savings of DVFS. In another work of Kanga [16], the author presents the implementation of the solution proposed before, focusing on the ondemand DVFS policy, making this policy suitable to virtualized environments. The paper analyzes the changes of frequency-dependent rate of processor utilization, and offers pre-defined limits that make greater energy savings based on the accuracy of these adjustments.

Differently from the related works, this work evaluates the energy consumption in virtualized environments, focusing on the trade-off between different DVFS policies and their impact on network applications.

IV. Experiments

This section presents the experiments performed, describing the testbed, benchmarks, DVFS settings, and network metrics used. Afterwards, the results obtained are presented and analyzed.

A. Testbed

Evaluations were performed on a client-server architecture, simulating a client node accessing to virtualized applications in a server node, connected by a Gigabit Ethernet network. The server used in our experiments consists of 2 Intel Xeon E5520 (16 cores in total), 2.27GHz, 16 Gb RAM. This server runs the Citrix XenServer, a well-known virtualization solution in industry. In each set of tests, DVFS was configured with three operating policies: *performance*, *ondemand*, and *powersave*. The energy consumption was obtained using a multimeter which is connected between the power source and the server. This device (EZ-735 digital multimeter) has a USB connection that allows periodic external reading and gives the values of power consumption in watts-per-hour.

The network performance metrics evaluated during the experiments were: throughput and jitter. Throughput is the value that indicates the effective data rate transfer per second, while jitter is the variation in delivery time of packets in a given space of time. This variation is directly related to the network demand. The evaluation of throughput focused on the impact in energy savings and response time to the user. The evaluation of jitter aimed at analyzing the impact of the virtualization layer in the variation of data packets delivery, which consequently impacts on energy waste.

The experiment architecture is described in Figure 3. The client part of the benchmark performs requests, using the network, to applications hosted on two distinct virtual machines. Each virtual machine is associated with one of the two processors available, forcing that changes in frequency of both processors can directly influence each virtual machine, and consequently, the application within each one of them.

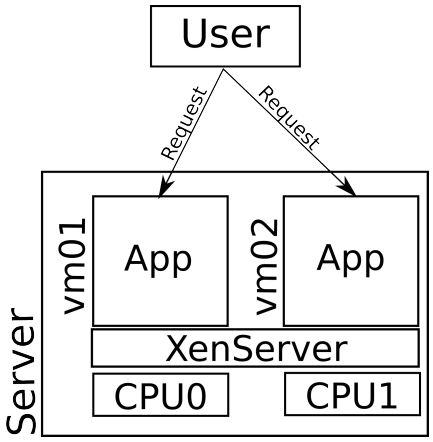


Figura 3. Experiment Architecture

Evaluations of the trade-off between the impact of changes in processor frequency and network throughput was evaluated

and monitored through the benchmarks: Hping [17], T50 [18], Apache-Bench [19] and Iperf [20].

The first benchmark used was Hping. This benchmark is a packet generator that is used to analyze TCP/IP protocols. Currently, in its 3rd version, hping is one of the standard tools for security auditing and testing of firewalls and networks. Hping is programmable using the Tcl language, which allows programmers to develop their own scripts for manipulation and analysis packages.

The second benchmark used was T50 Sukhoi PAK FA Mixed Packet Injector. This tool was developed for the purpose of packet injection, designed primarily to test DoS/DDoS attacks. From the basic use of stress testing, T50 is capable of sending requests as follows: a value higher than one million packets per second of SYN Flood (+50% of the uplink network) to a network 1000BASE-T (Gigabit Ethernet) and more than 120.000 packets per second of SYN Flood (+60% of the network uplink) in a 100BASE-TX (Fast Ethernet). Additionally, it can send Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) protocols sequentially (with only microseconds difference). It is licensed under the GPL version 2.0.

The third benchmark used was Apache-Bench. This benchmark can measure the Hypertext Transfer Protocol (HTTP) server performance, running concurrent requests, and is especially efficient for test environments where Apache runs on multicore. The metric to be evaluated consists of requests per second at a given time interval, allowing to visualize the impact of various hardware components on web server performance.

The last benchmark used was Iperf. This benchmark is used to test various network metrics such as bandwidth and jitter, which can perform packet injection (TCP and UDP) to measure the performance of these networks. This tool was developed by Distributed Applications Support Team (DAST) and the National Laboratory for Applied Network Research (NLANR), and it can run on many platforms, including Linux, Unix, and Windows.

B. Results

The first evaluation shown in Figure 4 presents the virtualized server performance to answer requests in a given time interval. The results show that performance and ondemand policies kept the 10000 requests, ending his run in a shorter time than powersave which managed to answer on average smaller requests. The ondemand policy takes a little more time to complete its execution when compared to the performance policy, as there is an overhead in setting the frequencies to the behavior of the application. The powersave policy behavior is an expected result because the processor frequency is limited to one lower than the other two policies.

Figure 5 shows that there is little difference in energy consumption between performance and ondemand policies. This happens according to the benchmark behavior, which always tries to keep the processing to the highest during the test period. Therefore, the frequency variation that enables

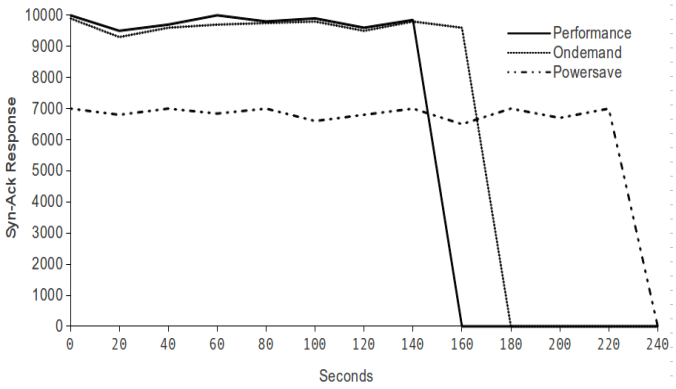


Figura 4. Hping Performance

ondemand policy is quite limited. A big difference could be seen in a case where there is a low rate of requests and, consequently, a low rate of processor utilization. However, there is a significant difference between these two DVFS policies and the powersave policy. Despite this policy save around 10% of energy, there is an increase in response time by 70%.

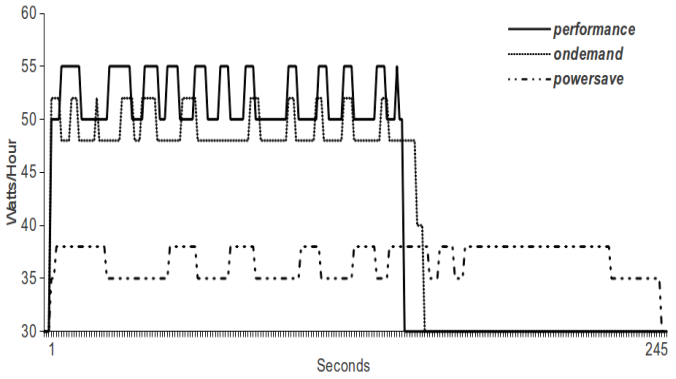


Figura 5. Hping Power Consumption

The second benchmark (T50) tested again the performance of the web server, through a flood of requests, trying to keep for a certain period of time, the most supported requests. The performance results can be seen in Figure 6. Performance and ondemand policies managed to keep the service in an average time of 150 seconds. Instead, the powersave policy was able to answer only an average between 6000 and 7000 requests over a period of about 68% higher.

The T50 benchmark shows similar results in power consumption behavior. These results can be seen in Figure 7. Again, there is no significant difference between the performance and ondemand policies. Regarding powersave policy, this enables energy savings of 15% when compared to the performance policy.

Tests using Apache-Bench perform requests to a real HTTP server. In this experiment, were performed a range between 100 and 1000 requests per second, evaluating how many milliseconds would lead the server to respond to all of them. Figure 8 shows the higher the number of requests, the greater

 ISBN: 978-1-61208-318-6

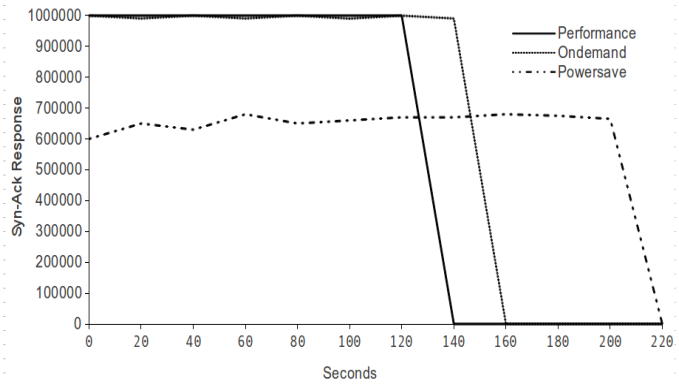


Figura 6. T50 Performance

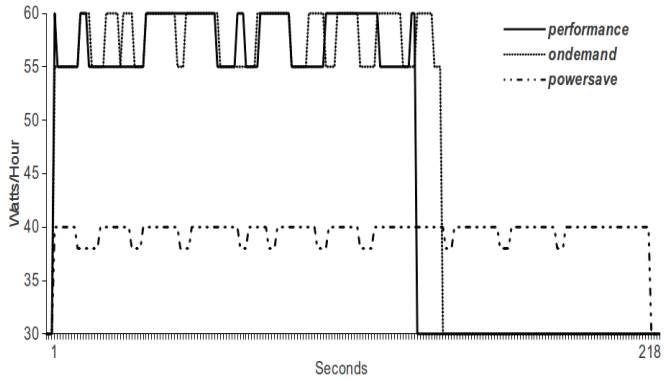


Figura 7. T50 Power Consumption

the response time in milliseconds. The ondemand policy is very near to the response times achieved by the performance policy. Both have a response time for all cases on average 35% faster than the powersave policy, which really shows that the frequency of the processor directly affects the network performance applications.

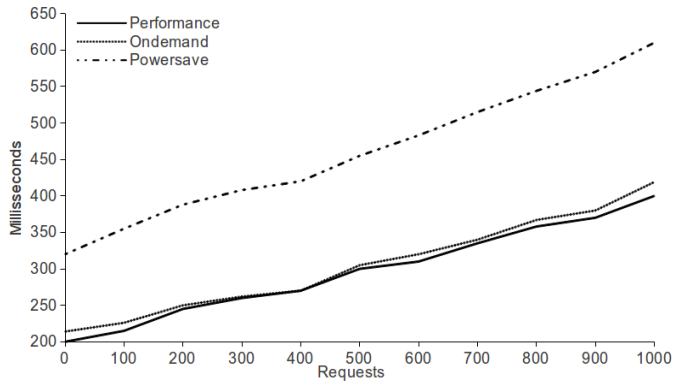


Figura 8. Apache-Bench Performance

Concerning power consumption, Figure 9 shows that performance and ondemand policies try to keep the highest processor utilization during the execution time of the application, to respond to the requests in the shortest time possible. With the limited frequency of the processor in powersave policy, there is much energy-saving, although its impact is significant on performance.

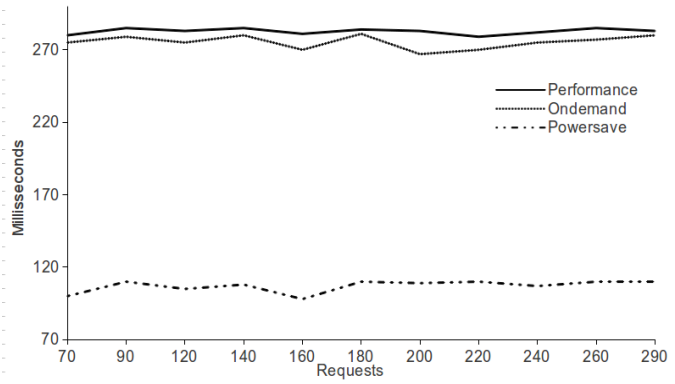


Figura 9. Apache-Bench Power Consumption

Figure 10 shows the jitter test. In these tests, DVFS policies from a native linux environment were compared to virtualized DVFS policies. The results showed that there are differences when comparing jitter on the environment in any of the native DVFS policies against a virtualized environment. Based on this, it can be verified that virtualized environments causes jitter overhead, which can cause an inefficient service for certain types of applications, such as video streaming. Furthermore, there is also a greater impact when using the powersave policy. This is probably due to the delay imposed by the structure of the ring buffer from Xen.

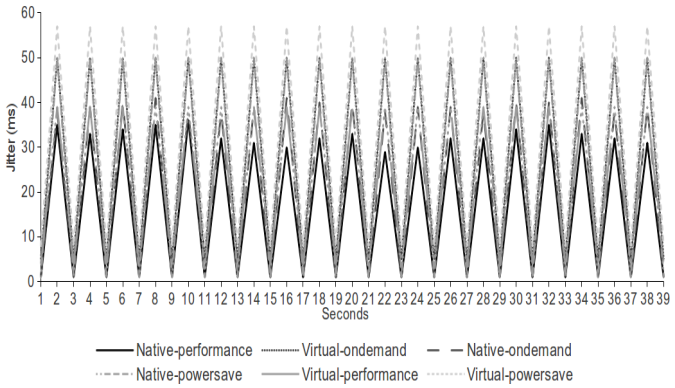


Figura 10. Iperf Jitter Evaluation

The evaluations performed allowed an examination on issues of QoS for virtualized networks. The QoS is defined in terms of the Service Level Agreements (SLA) with features such as the least throughput, maximum response time or latency time. A network architecture that can manage traffic dynamically according to SLAs is not only important for the future competitiveness, but can also set the basis for a systematic approach to energy efficiency. However, the implementation of QoS can actually increase the total network traffic and energy consumption of their virtualized environments.

The tests showed that by decreasing the bandwidth, latency increases. From the point of view of energy consumption, it is necessary to improve the latency by increasing the bandwidth, which directly impacts on energy consumption.

 ISBN: 978-1-61208-318-6

On this point, it must be dealt aspects such as component choice and consolidation of I/O. Likewise, it is necessary to investigate networks without loss in performance compared to the bandwidth and energy efficiency. For example, package lossless network protocols usually means more complex and more latency, as well as more processing power and low-bandwidth efficiency.

V. Conclusion and Future Work

In February 2007, the main leaders of the IT industry have announced The Green Grid, a nonprofit consortium whose mission is to improve the energy efficiency of data centers and business ecosystems based on computing. The strategy is to encourage the development of chips, servers, networks and other solutions that consume energy more efficiently.

Some of these efforts have focused on technologies such as virtualization. However, virtualization technology incurs in a processing overhead, through the addition of an abstraction layer that translates all requests between the virtual machine and physical host. This layer is affected by other technologies that attempt to promote energy-savings, such as DVFS.

This paper evaluated the impact of DVFS on network-dependent applications in virtualized environments, focusing on network performance. The choice of this metric is justified by the impact on response time for user applications. Furthermore, we also evaluated the overhead of the virtualization layer on jitter, a metric that can impact on energy waste, as well as quality of service.

As future work, we intend to evaluate changing some network parameters, such as the application buffer size, system buffer size, and Maximum Transmission Unit (MTU), that might influence in power consumption, as well as the influence of network throughput.

Referências

[1] G. Mone, "Redesigning the data center," vol. 55, no. 10. New York, NY, USA: ACM, Oct. 2012, pp. 14–16.

[2] R. Nathuji and K. Schwan, "Virtualpower: coordinated power management in virtualized enterprise systems," vol. 41, no. 6. New York, NY, USA: ACM, Oct. 2007, pp. 265–278.

[3] Q. Zhu, J. Zhu, and G. Agrawal, "Power-aware consolidation of scientific workflows in virtualized environments," in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1–12.

[4] C. Humphries and P. Ruth, "Towards power efficient consolidation and distribution of virtual machines," in *Proceedings of the 48th Annual Southeast Regional Conference*, ser. ACM SE '10. New York, NY, USA: ACM, 2010, pp. 75:1–75:6.

[5] V. Spiliopoulos, G. Keramidas, S. Kaxiras, and K. Efstathiou, "Poster: Dvfs management in real-processors," in *Proceedings of the international conference on Supercomputing*, ser. ICS '11. New York, NY, USA: ACM, 2011, pp. 373–373.

[6] H. Hanson, S. W. Keckler, S. Ghiasi, K. Rajamani, F. Rawson, and J. Rubio, "Thermal response to dvfs: analysis with an intel pentium m," in *Proceedings of the 2007 international symposium on Low power electronics and design*, ser. ISLPED '07. New York, NY, USA: ACM, 2007, pp. 219–224.

[7] G. Mateescu, "Overcoming the processor communication overhead in mpi applications," in *Proceedings of the 2007 spring simulation multiconference - Volume 2*, ser. SpringSim '07. San Diego, CA, USA: Society for Computer Simulation International, 2007, pp. 375–378.

[8] T. Brecht, G. J. Janakiraman, B. Lynn, V. Saletore, and Y. Turner, "Evaluating network processing efficiency with processor partitioning and asynchronous i/o," in *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, ser. EuroSys '06. New York, NY, USA: ACM, 2006, pp. 265–278.

[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," vol. 37, no. 5. New York, NY, USA: ACM, Oct. 2003, pp. 164–177.

[10] I. Takouna, W. Dawoud, and C. Meinel, "Energy efficient scheduling of hpc-jobs on virtualize clusters using host and vm dynamic configuration," vol. 46, no. 2. New York, NY, USA: ACM, Jul. 2012, pp. 19–27.

[11] D. G. d. Lago, E. R. M. Madeira, and L. F. Bittencourt, "Power-aware virtual machine scheduling on clouds using active cooling control and dvfs," in *Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science*, ser. MGC '11. New York, NY, USA: ACM, 2011, pp. 2:1–2:6.

[12] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831.

[13] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," vol. 41, no. 1. New York, NY, USA: John Wiley & Sons, Inc., Jan. 2011, pp. 23–50.

[14] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831.

[15] C. M. Kamga, G. S. Tran, and L. Broto, "Power-aware scheduler for virtualized systems," in *Green Computing Middleware on Proceedings of the 2nd International Workshop*, ser. GCM '11. New York, NY, USA: ACM, 2011, pp. 5:1–5:6.

[16] C. M. Kamga, "Cpu frequency emulation based on dvfs," in *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, ser. UCC '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 367–374.

[17] S. Sanfilippo. (2013, Oct.) Hping. [Online]. Available: http://www.hping.org

[18] N. Brito. (2013, Oct.) T50. [Online]. Available: http://t50.sourceforge.net

[19] A. Foundation. (2013, Oct.) Apache-bench. [Online]. Available: http://httpd.apache.org/docs/2.4/programs/ab.html

[20] NLANR/DAST. (2013, Oct.) Iperf. [Online]. Available: http://iperf.sourceforge.net

 ISBN: 978-1-61208-318-6