
# ICNS 2019

The Fifteenth International Conference on Networking and Services

ISBN: 978-1-61208-711-5

June 2 - 6, 2019

Athens, Greece

**ICNS 2019 Editors**

Kiran Makhijani, Futurewei Technologies, USA

# ICNS 2019

# Forward

The Fifteenth International Conference on Networking and Services (ICNS 2019), held between June 02, 2019 to June 06, 2019 - Athens, Greece, continued a series of events targeting general networking and services aspects in multi-technologies environments. The conference covered fundamentals on networking and services, and highlights new challenging industrial and research topics. Ubiquitous services, next generation networks, inter-provider quality of service, GRID networks and services, and emergency services and disaster recovery were considered.

IPv6, the Next Generation of the Internet Protocol, has seen over the past three years tremendous activity related to its development, implementation and deployment. Its importance is unequivocally recognized by research organizations, businesses and governments worldwide. To maintain global competitiveness, governments are mandating, encouraging or actively supporting the adoption of IPv6 to prepare their respective economies for the future communication infrastructures. In the United States, government's plans to migrate to IPv6 has stimulated significant interest in the technology and accelerated the adoption process. Business organizations are also increasingly mindful of the IPv4 address space depletion and see within IPv6 a way to solve pressing technical problems. At the same time IPv6 technology continues to evolve beyond IPv4 capabilities. Communications equipment manufacturers and applications developers are actively integrating IPv6 in their products based on market demands.

IPv6 creates opportunities for new and more scalable IP based services while representing a fertile and growing area of research and technology innovation. The efforts of successful research projects, progressive service providers deploying IPv6 services and enterprises led to a significant body of knowledge and expertise. It is the goal of this workshop to facilitate the dissemination and exchange of technology and deployment related information, to provide a forum where academia and industry can share ideas and experiences in this field that could accelerate the adoption of IPv6. The workshop brings together IPv6 research and deployment experts that will share their work. The audience will hear the latest technological updates and will be provided with examples of successful IPv6 deployments; it will be offered an opportunity to learn what to expect from IPv6 and how to prepare for it.

Packet Dynamics refers broadly to measurements, theory and/or models that describe the time evolution and the associated attributes of packets, flows or streams of packets in a network. Factors impacting packet dynamics include cross traffic, architectures of intermediate nodes (e.g., routers, gateways, and firewalls), complex interaction of hardware resources and protocols at various levels, as well as implementations that often involve competing and conflicting requirements.

Parameters such as packet reordering, delay, jitter and loss that characterize the delivery of packet streams are at times highly correlated. Load-balancing at an intermediate node may, for example, result in out-of-order arrivals and excessive jitter, and network congestion may

manifest as packet losses or large jitter. Out-of-order arrivals, losses, and jitter in turn may lead to unnecessary retransmissions in TCP or loss of voice quality in VoIP.

With the growth of the Internet in size, speed and traffic volume, understanding the impact of underlying network resources and protocols on packet delivery and application performance has assumed a critical importance. Measurements and models explaining the variation and interdependence of delivery characteristics are crucial not only for efficient operation of networks and network diagnosis, but also for developing solutions for future networks.

Local and global scheduling and heavy resource sharing are main features carried by Grid networks. Grids offer a uniform interface to a distributed collection of heterogeneous computational, storage and network resources. Most current operational Grids are dedicated to a limited set of computationally and/or data intensive scientific problems.

Optical burst switching enables these features while offering the necessary network flexibility demanded by future Grid applications. Currently ongoing research and achievements refers to high performance and computability in Grid networks. However, the communication and computation mechanisms for Grid applications require further development, deployment and validation.

We welcomed academic, research and industry contributions. The conference had the following tracks:

- NGNUS: Next Generation Networks and Ubiquitous Services
- CGNS: CLOUD/GRID Networks and Services
- COMAN: Network Control and Management

We take here the opportunity to warmly thank all the members of the ICNS 2019 technical program committee, as well as all the 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 effort to contribute to ICNS 2019. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the ICNS 2019 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that ICNS 2019 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the areas networking and services. We also hope that Athens, Greece provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

**ICNS 2019 Chairs**

**ICNS Steering Committee**

Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Mary Luz Mouronte López, Universidad Francisco de Vitoria - Madrid, Spain
Massimo Villari, Universita' di Messina, Italy
Éric Renault, Institut Mines-Télécom - Télécom SudParis, France

**ICNS 2019**
**Committee**

**ICNS Steering Committee**
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Mary Luz Mouronte López, Universidad Francisco de Vitoria - Madrid, Spain
Massimo Villari, Universita' di Messina, Italy
Éric Renault, Institut Mines-Télécom - Télécom SudParis, France
Robert Bestak, Czech Technical University in Prague, Czech Republic
Young-Joo Suh, POSTECH (Pohang University of Science and Technology), Korea
Gledson Elias, Federal University of Paraíba (UFPB), Brazil
Rui L.A. Aguiar, University of Aveiro, Portugal
Ivan Ganchev, University of Limerick, Ireland / Plovdiv University "Paisii Hilendarski", Bulgaria

**ICNS Industry/Research Advisory Committee**
Steffen Fries, Siemens, Germany
Alex Sim, Lawrence Berkeley National Laboratory, USA
Jeff Sedayao, Intel Corporation, USA
Juraj Giertl, T-Systems, Slovakia

**ICNS 2019 Technical Program Committee**

Wessam Afifi, Mavenir Systems in Richardson, USA
Rui L.A. Aguiar, University of Aveiro, Portugal
Pouyan Ahmadi, George Mason University, USA
Mehmet Aksit, University of Twente, Netherlands
Markus Aleksy, ABB AG, Germany
Alexandros Apostolos Boulogeorgos, Aristotle University of Thessaloniki, Greece
Patrick Appiah-Kubi, University of Maryland University College, USA
Mohammad M. Banat, Jordan University of Science and Technology, Jordan
Ilija Basicevic, University of Novi Sad, Serbia
Meriem Kassar Ben Jemaa, National Engineering School of Tunis (ENIT), Tunisia
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Jihen Bennaceur, École nationale des sciences de l'informatique, Tunisia
Juan Carlos Bennett, SPAWAR Systems Center Pacific, USA
Luis Bernardo, Universidade Nova de Lisboa, Portugal
Robert Bestak, Czech Technical University in Prague, Czech Republic
Eugen Borcoci, University "Politehnica" of Bucharest (UPB), Romania
Fernando Boronat Seguí, Universidad Politécnica De Valencia-Campus De Gandia, Spain
Safdar Hussain Bouk, Kyungpook National University, Daegu, Republic of Korea
Christos Bouras, University of Patras / Computer Technology Institute and Press - Diophantus,

Zsolt Polgar, Technical University of Cluj Napoca, Romania
Ziad Qais, The University of Manchester, UK
Md Arafatur Rahman, University Malaysia Pahang, Malaysia
Mayank Raj, IBM, USA
Adib Rastegarnia, Purdue University, USA
Da Qi Ren, Futurewei Technologies Inc., USA
Éric Renault, Institut Mines-Télécom - Télécom SudParis, France
Sebastian Robitzsch, InterDigital Europe, UK
Ignacio Sanchez, University of the West of Scotland, UK
Nico Saputro, Florida International University, USA / Parahyangan Catholic University, Indonesia
Panagiotis Sarigiannidis, University of Western Macedonia, Greece
Jeff Sedayao, Intel Corporation, USA
Purav Shah, Middlesex University, UK
Alireza Shams Shafigh, University of Oulu, Finland
Anas Shatnawi, LIP6 - Sorbonne University, Paris, France
Alex Sim, Lawrence Berkeley National Laboratory, USA
Vasco N. G. J. Soares, Instituto de Telecomunicações / Instituto Politécnico de Castelo Branco, Portugal
José Soler, DTU Fotonik, Denmark
Karthikeyan Subramaniam, Samsung R & D Institute, Bangalore, India
Young-Joo Suh, POSTECH (Pohang University of Science and Technology), Korea
Yoshiaki Taniguchi, Kindai University, Japan
Giorgio Terracina, Università della Calabria, Italy
Ishan Vaishnavi, Huawei Technologies, Munich, Germany
Hans van den Berg, TNO / University of Twente, Netherlands
Ioannis Vardiambasis, Technological Educational Institute (TEI) of Crete, Greece
Vladimir Vesely, Brno University of Technology, Czech Republic
Quoc-Tuan Vien, Middlesex University, UK
Massimo Villari, Universita' di Messina, Italy
Ferdinand von Tüllenburg, Salzburg Research Advanced Networking Center, Austria
Jin-Yuan Wang, Peter Grünberg Research Center | Nanjing University of Posts and Telecommunications, China
Junwei Wang, University of Hong Kong, Hong Kong
Mingkui Wei, Sam Houston State University, USA
Michelle Wetterwald, HeNetBot, France
Cong-Cong Xing, Nicholls State University, USA
Anjulata Yadav, Shri G.S. Institution of Technology and Science, Indore, India
Sherali Zeadally, University of Kentucky, USA
Ning Zhang, Texas A&M University at Corpus Christi, USA
Tao Zheng, Orange Labs China, China
Jiazhen Zhou, University of Wisconsin – Whitewater, USA
Ye Zhu, Cleveland State University, USA
Taieb Znati, University of Pittsburgh, USA

**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**

# Application of Human Profiling by Agent for Activating Human Communication

Masafumi Katoh, Junichi Suga, Yuji Kojima, Masaaki Kawai
Service Centric Network Research Center
Fujitsu Laboratories Ltd.
Kamikodanaka 4-1-1, Nakahara-ku, Kawasaki 211-8588, Japan
email:{katou.masafumi, suga.junichi, kojima.yuuji, kawai.masaaki}@fujitsu.com

***Abstract*—In this paper, we propose to put an agent in the network to help activate human communication. Our hypothesis is that human behavior depends both on a static profile and a dynamic context. So, we verify our hypothesis by performing repeated experiments. In our experiments, one author acted as the agent and collected many responses from members in an organization. As a result, an effective messaging style can be found by understanding their profile. Next, we categorize their profiles into stubborn one and flexible one. A small amount of data could not make any impact on the members with the stubborn profile, but made an impact on those with the flexible profile, depending on their context. So, we confirm that the agent should understand the profile and context of the object persons, and transform the data to effectively convey the client's intention. Finally, we address a design method of human profiling agent.**

***Keywords- context; profile; agent; human communication; message; CPS.***

## I. INTRODUCTION

People spend a lot of time in the cyber world in business and daily life because of the popularity in World Wide Web services [1][2]. Combining sensor data with the Web by Internet of Things (IoT) [3], data in the physical world are applied for various context-aware applications [4]. Even a hand gesture can be interpreted in cyber world with the advanced wearable devices [5]. Such trend suggests a Cyber-Physical System (CPS) will gradually be positioned, in which various context data can be handled.

However, we often feel frustrated in communication. Reference [6] broadly defines communication as all procedures by which one's mind affects another. The authors have categorized problems of communication into three levels: i) accuracy of transmitted symbol, ii) preciseness of conveyed symbol, and iii) effectiveness of received meaning to lead the receiver's conducts. Our objective is the 3rd level, that is, to build networking environment where the information sender can effectively transmit his/her intention, emotion, dearness and so on.

On the other hand, problems for the receiver are predominantly caused by the imbalance between the human perception and the volume of input data. It is known that the average human ability to percept information through eye or ear is limited to 223 bps or 105 bps, respectively [7]. So, excessive data will interfere in the human cognition, and will frustrate people, as follows.

A) People take a lot of time mining data with value.
B) People receive many data that could miss the mark. For example, although they hit the category like classical music, they never hit the favorite composer in a short time.
C) People receive data when it is not useful. For example, the announcement of the disruption of the commuting train service just before they arrive at the transit station is too late.

In summary, a data sender wants to convey his/her intent to a receiver. Then, we envision that a software agent mediates between them through the network since the agent could quickly process a lot of data without vital limitation or emotional barrier.

We will model the human agent and verify its effect. In Section II, we briefly review the trend of past studies about context awareness and an agent as the related works. In Section III, we show our hypothesis on human behavior that is based on the static profile and the dynamic context of object persons [8]. In Section IV, we describe the model of the proposed human agent and its role. In this paper, the human agent aims to activate the receiver's reaction by modifying the input data from the client. In Section V, we evaluate the effectiveness to figure out the profile and the context to activate the reaction of the object persons through repeated experiments. In our experiments, an author of this paper acts as the agent in place of the software program. In Section VI, we discuss how to design the human agent. Here, the design procedures in our experiment are categorized into those that are programed by designers and those that are automatically implemented as algorithm by the agent.

## II. RELATED WORKS

Many studies on context awareness have been performed in the past decade [9]. Typical use scenarios are a dynamic resource allocation by referring to user status and network environment [10] [11]. Other scenarios are service control, such as the screen structure of a mobile terminal that is switched depending on the user's location [12]. Most studies have assumed that the role of the context information could be common to everybody. However, the significance of

 ISBN: 978-1-61208-711-5 

context information must be different to each person. So, we want to coordinate the context and the human profile.

We proposed to have an agent assist the data process in place of person. Actually, a lot of studies on the agent have already been done from various viewpoints [13]. The typical applications of agent is personal assistance [14][15] and the agent provides the means of a specific issue. For example, an agent categorizing data was proposed for personal data market [16]. A personal agent mediating personal knowledge management like transformation between tacit and explicit knowledge was proposed too [17]. The design of the multi-agent has already begun to be studied and the agent's ontology model was proposed [18][19]. Further, remarking the penetration of Artificial Intelligence (AI) speaker or personal agent in mobile phones, it is clear that a conversation between a person and AI has progressed [20][21].

Our viewpoint is different from others since we assume three kinds of stakeholders, i.e., a client, object persons, and 3rd parties. Our agent assists the client to preferentially convey his/her messages to the object persons while referring to the profile of the object persons and the context around them.

## III. Requirements and Hypothisis

### A. General requirements for human communication

Let us describe our approach to activate human communication. Problem A) in Section I will be resolved by searching valuable data and filtering out trivial data by an agent understanding human profile such as concern or taste. Problem B) can be improved by matching between the input data and the receiver's profile. Problem C) is caused by missing the receiver's timely requests. This issue will be resolved by understanding his/her external context. Repeatedly, the value of input data is varied depending on various elements such as the personality, mind, timing and place. Therefore, we have proposed to put an agent in the network, which figures out the profile and the context of the object persons [8].

### B. Hypothesis and terminologies

Let us discuss factors that affect a person who may react. We divide the impact factors into a static human profile and a dynamic human context, as shown in Table I and Figure 1.

A person grows through the experience with sensory information and language [22]. Then, the personality has been formed for a long period and it is relatively static. The personality is thought the principle how to feel, think and act, and cannot easily be changed by a small number of messages. We call the abstraction of the personality a human profile (profile from now on) which a third party characterizes through the observation.

On the other hand, we call the dynamic elements a human context (context from now on). Further, we categorize the contexts into 3 elements: (1) internal context, (2) external context and (3) data input [8][23]. An internal context is the internal state of a person such as mind, an emotion and a vital condition. It is thought to be the dynamic part around the profile and can be changed even by a small number of messages. An external context is an external state around the person such as time, place, company and belongings. Data input is a kind of change of the external context. However, we conveniently divide the data input from the external context since the data is an object which Information and Communication Technologies (ICT) can process, even though the data is originally generated based on the sender's intention. Further, the person cannot control most of the external context like time. However, the person can switch the various external contexts by acting such as talking with colleagues, writing a document, walking or running, etc.

TABLE I. IMPACT FACTORES ON HUMAN BEHAVIOR

| | Elements | Examples of element affecting person |
|---|---|---|
| Static | Profile | Full name, Mother language, Gender, Age, Contact address, Address, Career and title, Financial resource or borrowing, Record of health/illness |
| Dynamic | Internal Context | Will, Desire, Mind, Emotion, Physical condition |
| | External Context | Time, Place, Accompanies, Belongings, Social events, Natural phenomenon |
| | Data input | Dialog, Chat, E-mail, Phone call, SNS, News, Papers, Books |

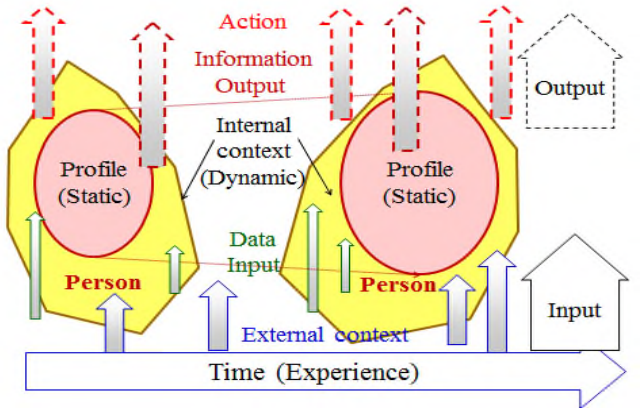



Figure 1. Hypothesis about impact factors on human reactions.

The horizontal line of Figure 1 represents time. The person encounters various external contexts including data input. Although the profile grows gradually, the internal context is dynamically moved by the change of external context. As a result, the person will react by talking with friends, writing a document, expressing emotion, and going somewhere.

 ISBN: 978-1-61208-711-5 

## IV. PROPOSAL OF HUMAN AGENT

### A. Proposed human agent for activating communication

We propose to introduce a human agent between a sender of information and a receiver of data. The agent prompts receiver's reaction according to the intention of its client [23]. Here, a client is defined as someone who specifies the role of the agent. Not only the object person oneself, but also somebody else could be the client. For example, the object person oneself may hire the agent as his/her secretary, and his/her parents may hire the agent as a tutor of their child. It is noted that the client must have a piece of the profile because they have already accompanied the object person.

Figure 2 shows the basic model of the human agent between the client and the object persons. The agent tries to make the lively reaction of the object persons by moving their internal context. It temporarily keeps input data which the client sends, and transforms the data so as to meet the client's intention. At that time, the agent refers the profile of the object persons. It is noted that the profile is just an abstraction of the personality through observation. So, the profile is not necessarily true. Further, it is not fixed forever, since the person grows. Therefore, although a designer initially sets the profile which is told by the client, the agent should update it through the observation of the persons' actual reaction. So, the agent must have a feedback mechanism which monitors the information output or reactions to the data input because it is difficult to directly measure the internal context.

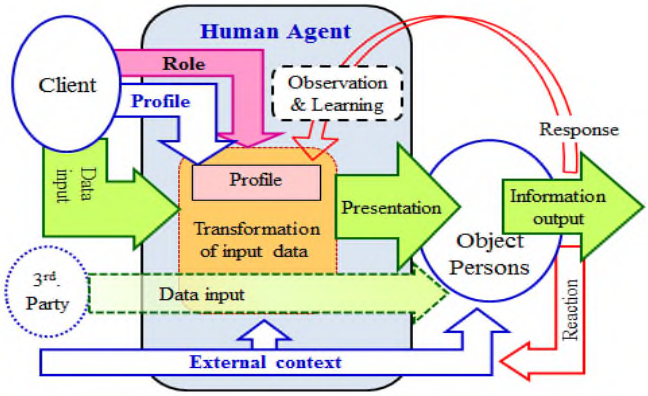



Figure 2. Model of human agent for activating communication.

Currently, we position the agent as an application program to communicate with the client, the object persons and a huge number of third parties, as in Figure 2. Any computing resource is acceptable to carry on the agent program, such as terminal equipment and the node in the Internet or a data center as long as it can communicate with the above stakeholders. If we were to use one word, the entity for the agent should be located at the center among the stakeholders to save network resource [8].

### B. Transformation of input data from the client

This paper focuses on the agent to convey the intention of the client to the object persons, even though the object person must be frustrated due to the excessive data of little interest. We assume that the agent is prohibited to abandon or change any data from the third parties since it could be significant to the receiver. What the agent can do is to transform the input data from the client.

Table II is the alternatives of transformation to affect the person. To senior people, for example, message with high volume voice can easily be understood, but small size letters are never welcome. It cannot be doubted that *A*, *B* or *C* in Table II are effective depending on their profile such as age and mother tongue. Therefore, we will evaluate the effect of *D* and *E* in the remaining sections.

TABLE II. POSSIBLE TRANSFORMATIONS OF INPUT DATA BY AGENT

| Transformed objects | Example of transformations |
|---|---|
| *A) Language* | Translate message to mother language |
| *B) Displayed form* | Select displayed form such as voice/text/graph/picture/animation/video |
| *C) Size or volume* | Adjust letter size or audio volume |
| *D) Message style* | Select message style and arrange timing to send |
| *E) Expression or rhetoric* | Change tone of words such as order, modest request, or heartstrings |

### C. Metric of human agent for positive response

Let us assume a scenario where a client wants to know lively opinions of members in his/her organization. Then, the client hires the human agent profiling a set of object members. The agent tries to have the members respond as much as possible by transforming the request message for the opinion survey. The mere repeat of requests must be annoying and ineffective since the human recognition capability has a limit. So, the agent must find a smart presentation style of the data from the client without increasing the amount of data. Considering the above, we define "response rate" as a metric for activating their level.

- Response rate ($r$) of a person is expressed by $n_o/n_i$
  $n_o$: the number of output information, i.e., response
  $n_i$: the number of input data, i.e., request for response
- Response rate ($R$) of a set of object persons is expressed by $\Sigma\ (n_o/n_i)\ p(n_o, n_i) / N$.
  $p(n_o, n_i)$: the number of members such that $n_o$ responses to $n_i$ requests. Here, N is the total number of object persons, i.e., examinees in later.

The human agent cuts and tries to transform the request message from the client to collect many responses while observing the reactions of the object persons.

 ISBN: 978-1-61208-711-5 

## V. EXPRIMENTS

### A. Method of agent's simulation

Considering that a software agent is developed by designers, it cannot execute logical judgement beyond that of the designers. In our experiments, an author, M. Katoh, acts as the agent in place of the designer since he knows the clients profiles and recognizes their context. Next, we analyze our procedure from the view point of the feasibility by software. The purposes of the experiments are as follows.

1. Verifying hypothesis that the profile of the object person is essential to move him/her in experiment I.
2. Verifying hypothesis that the context of the object person is important to activate him/her in experiment II.
3. Addressing how to design the agent which refers to the profile and the external context of the object person.

Instead of the software, Katoh requested to return examinees' opinions as much as possible by e-mail. The number of examinees "N" is 42 or 30 in the experiments I or II, respectively. The question form includes 3 or 4 choices for answer so that they will respond for a few minutes. The deadline of response was for 48 hours so that they have couple of chances to check their received e-mails.

For experiment I, Katoh modified the message style and expression, and observed responses from 42 examinees to find out an effective presentation style. Table III summarizes the presentation styles for 7 trials. One request mail includes one question and choices for answer as indicated in APPENDIX. Here, we took care that the content of the question does not impact their reaction since we want to observe the impact by message style or expression. That is, the questions should be nearly equal to interest them in each trial. So, considering that all examinees are researchers on ICT, he asked them about the high-level view about ICT.

TABLE III. PRESENTATION FOR EXPERIMENT I
(Examinees N=42)

| Trial) Date | *D) Message style* | *E) Rhetoric (Naming, Additional data)* |
|---|---|---|
| #1) 1/24 | (I) | ① Minimum as standard |
| #2) 1/29 | Requset: 1 to N | ② ①+Result of #1 |
| #3) 2/1 | (multicast by mailing list) | ③ Individual name + Result of #2 |
| #4) 2/6 | Response: 1 to 1 | ④ ③+Confidence policy + Result of #3 |
| #5) 2/14 | (II) | ⑤ Individual name |
| #5.5) | Request: 1 to 1 | ⑥ Remind for #5 |
| #6) 2/20 | Response: 1 to 1 | ⑦ ⑤ + Result of #5 |
| #7) 2/27 | (III) Request:1 to 6 groups Response: 1 to group (sharing in the group) | ⑧ Division to 6 groups. What is your group's choice? |

Here, the presentation style such as *D) message style* and *E) rhetoric* was changed for the series of trials. Figure 3 shows 3 message styles. The style (I) is that the request was sent in multicast (1 to N) by using the mailing list. Each examinee returns his/her answer to Katoh's address (1 to 1). The style (II) is a normal 1 to 1 communication. In the styles (I) and (II), all responses were gathered by only Katoh. To avoid this moral hazard due to the asymmetry of information, the style (III) adopts a mesh type communication (n to n). Here, 42 examinees were divided into 6 groups, and they shared his/her response in the group.

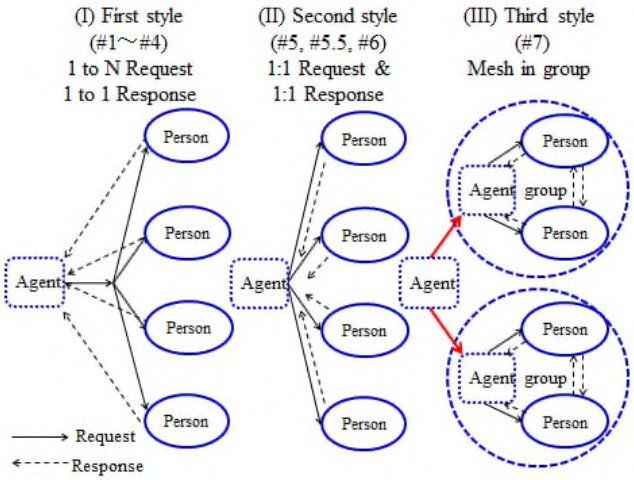



Figure 3. Message styles for experiment I.

TABLE IV. REQUEST TIMING FOR EXPERIMENT II
(Examinees N=30)

| Traial) Date | *D) Message style* | | |
|---|---|---|---|
| | *Style* | *Request timing* | |
| #1) 5/24 | (II) Request: 1 to 1 Response: 1 to 1 | Visual check | Absence |
| #2) 5/29 | | | Presence |
| #3) 6/4 | | Due to open scheduler in the organization | Before meeting |
| #4) 6/14 | | | During meeting |
| #5) 6/21 | | | During meeting |
| #6) 7/17 | | | Available |
| #7) 8/1 | | Visual check | Presence |
| #8) 8/7 | | | Presence |

In experiment II, Katoh sent the request mail for an opinion survey to 30 examinees. In this experiment, the message style was fixedly (II) since we wanted to find out not an effective style, but an effective context. Table IV summarizes requesting timings for 8 trials of this opinion survey. One mail includes one question and 3 or 4 options for answer, as in APPENDIX. Again, the contents for question should equally be interesting to them. So, he asked them about "working style" as a popular issue in Japan.

 ISBN: 978-1-61208-711-5

### B. *Results of experiement I*

Figure 4 shows the results of 8 trials during about 6 weeks. The blue bars represent the number of responses, and the red broken line represents the response rate ($R$).

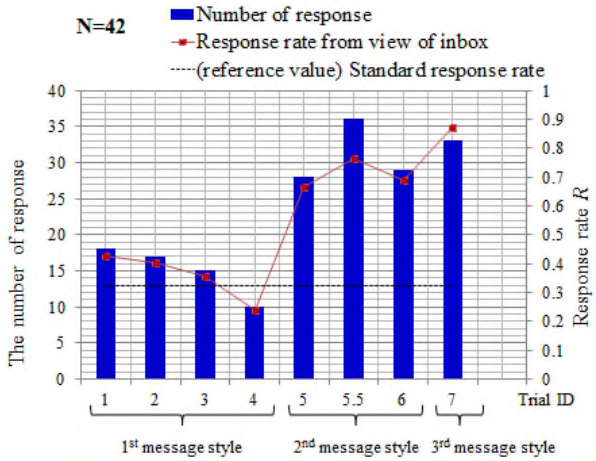



Figure 4. Transition of reactions in the experiment I.

In the message style (I) using mailing list for 42 examinees, the number of responding examines declined from 18 in the trial #1 to 10 in the trial #4. Although their curiosity seemed being stimulated at the trials #1 or #2, they may have gotten fatigued after trial #3. According to reports of opinion surveys for business persons, more than 2,300 in Japan, the average number of received and transmitted e-mails per day are about 40 and 13, respectively [24]. So, the average rate of information output to data input is 13/40 = 0.32 as a standard response rate. $R$ in the message style (I) declined from 0.42 to 0.24, which is lower than the standard response rate. During this period, the rhetoric was devised with an addition of the result for the previous trial, as in Table III. However, we could not observe the effectiveness at all. Rather, we observed that 62% of responses were returned within 2 hours. About 10 new e-mails pile in one's inbox during a meeting of 2 hours. So, the most valuable e-mails must first be responded after returning to his/her desk. As a result, the priority of the voluntary reaction becomes lower than secondary. That is, the effectiveness of our request is gradually weakened and lost for a couple of hours.

In the message style (II), Katoh sent 42 request mails to 42 examinees by using individual addresses. Further, the rhetoric is changed as in Table III. For example, the addressing style changed from "Hi, everybody" to "Addressing by individual full name". The number of responses of the trials #5 and #6 are 28 and 29, respectively. It can be recovered in a V-formation, and $R$ is over 0.67. So, we conclude that individual name and mail address play an important profile to get many responses. In the trial #5.5, he sent explicit reminder e-mails to 14 examinees having not responded yet, and then received 8 responses. It is noted that the response rate 8/14 of the second request for reminder is slightly lower than that of the initial response rate of 28/42. This fact means the request for only cool persons for the voluntary activity is lower than the average.

In the message style (III), we divided the 42 examinees into 6 groups according to the actual project in the organization. Katoh sent request mails to 6 groups and asked to share the response in the group. This style was effective to stimulate each other in the group. The number of responses became 33 and the response rate $R$ became the best, 0.87. The first reason is due to the effectiveness of one request. Since one request mail is shared for 7 examinees, the probability that 7 examinees are incognizant can be reduced. The second reason is due to the reminder being sent several times. That is, someone's response is shared, and so it can play a role of reminding others that they have not responded yet. That is, an original request will repeatedly be valid. So, we conclude that the organization structure as their profile plays an important role to get many responses. In fact, the response times were slightly more distributed than others.

### C. *Results of experiment II*

Figure 5 shows the results of 8 trials during about 10 weeks. Katoh sent the request mail to return their opinions by the message style (II). The blue bars represent the number of responses, and the red broken line represents the response rate $R$. In the trials #1 and #2, 26 examinees have responded, and $R$ is unexpectedly high 0.87. So, we had to try to reduce $R$ by finding out an inconvenient time due to their schedule management system. That is, for the trials #3, #4 and #5, he sent the request mail just before or during the meeting time. As a result, the number of responses was reduced to 21 or 22, and $R$ became lower to 0.7. We succeeded to reduce the number of responses, but it was not so dynamic. We think that the request timings in Table IV are all during working hours, so there is no drastic change of the external context. We got just 20 responses in the trial #6 even though we sent request mails at an available time according to the calendar. We think that the calendar is not necessary true to express their actual availability.

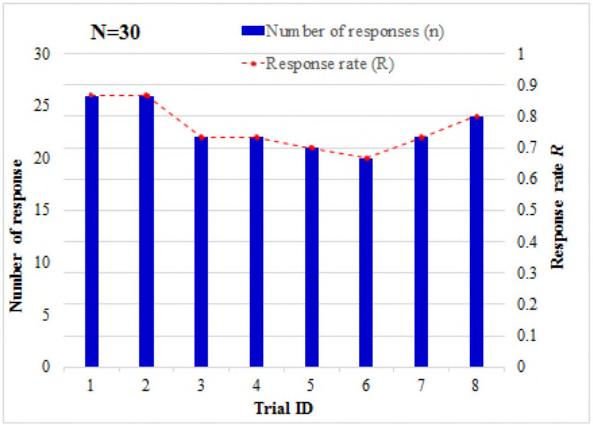



Figure 5. Transition of reactions in the experiment II.

It is sure that the variation of response of Figure 4 is more dynamic than that of Figure 5. That is, the mail address of the request message is the most dominant factor to get a positive response even though the request timing slightly impacts their reaction.

Figure 6 shows the examinees' characteristics. The horizontal line is the number of responses for 8 trials and the vertical line is the number of examinees. We can categorize 30 examinees into 3 groups. Group *A* is a set of positive 15 (50%) examinees who responded to all requests. Conversely, group *B* is a set of cool 2 (6.7%) examinees who never responded to any requests by voluntary cooperation. We define their profile as "stubborn", which strongly dominates their reaction. In other words, a small number of messages cannot impact their reaction.

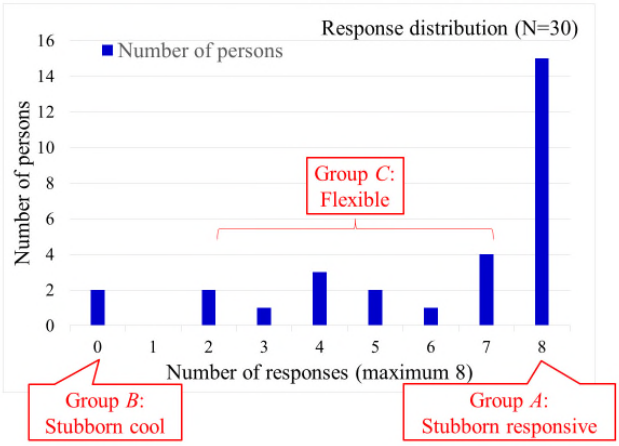



Figure 6. Characteristics of examinees and the category of profile.

Group *C* is a set of "flexible" 13 examinees who responded 2~7 times. Such reaction depends on external context. That is, if they receive the request when they are available, they respond for a couple of hours. Nevertheless, they lose the chance to return their opinions. Therefore, we conclude that the response of our questionnaire dominantly depends on the stubborn profile. Next, the reaction of flexible examinees depends on their availability, that is, their context.

## VI. CONSIDERATIONS

### A. Verification of our hypotheisis

➢ *Effectiveness of predetermined-profile*

Static profile such as individual name, email address, etc., can be pre-determined. It was clear that such profile played an important role to adopt the messaging style (II) to activate reactions of object persons. Further, by knowing the structure of the organization, the agent was able to adopt the message style (III), and got very high *R*, such as 0.85. If the client did not respond, the agent would have to try the possible arrangement of 6 groups from 42 persons. Considering that there are $42!/(7!)^6 = 8.57\text{x}10^{28}$ combinations, such cut and try approach is never feasible.

➢ *Effectiveness of learned-profile*

Through two experiments, we learned examinees personality, and were able to roughly predict the response rate *R*. Table V shows the result of the additional trial using the message style (III) for 3 groups consisting of 7 examinees. Before this trail, we averaged past response rate in experiment I for each examinee (in the second column of Table V). The third column shows our expectation of the number of responses in the group, in which Bernoulli trails are assumed. The right columns are the results, which are close to our expectation.

TABLE V. EXPECTED RESPONSE NUMBER BASED ON PAST RESULTS

| Group | Average response rate in the group | Our expectation based on Bernoulli trial | Result |
|---|---|---|---|
| *X* | 0.142 | 0.99 | 1 |
| *Y* | 0.751 | 4.0 | 6 |
| *Z* | 0.768 | 5.38 | 5 |

That is, the agent can roughly predict *R* by learning the profile through a number of trials, and brush up the profile of object persons. The group *Y* was more active than our expectation based on Bernoulli trials. This means that the reaction in group *Y* is more active than an independent trial, as described in Section V-B.

➢ *Effectiveness of context awareness*

We observed that examinees' reactions depend on their external context, as described in Section V-C. Especially, the request timing is crucial in order for flexible persons to be aware of the client's message and to react.

### B. Design Methodology

Although an author played the role of an agent for experiments, the agent must be described as a software program. Then, a key question is whether our cut and try approach in Section V can be described as a software. So, we classify the procedures into the manually coded ones and the algorithm based one. The former must be developed by designers based on the intent of the client, and the latter can automatically be executed by the agent. Figure 7 shows our current view. The dashed arrows represent the message flow by the human agent.

➢ *Manually coded procedures by designer*

A designer should define the role of the agent based on the client's requirement. The designer specifies the measurable metric to evaluate the effectiveness of the operation by the agent. Simultaneously, the client tells the designer the predetermined profile of the object persons. Next, the designer must set the possible methods which the agent can choose. In our experiments, alternatives such as 3 message styles, rhetoric transformation and request timing are set prior to the actual operation.

 ISBN: 978-1-61208-711-5 

➢ *Automatic procedure by algorithm in the agent*

In the online operation, the human agent dynamically chooses one method among pre-set alternatives for data input from the client to effectively transfer the client's intention. In the procedure, the agent refers the profile and the context of object person. Concerning rhetoric transformation, it is feasible to replace some words into stylized words based on a rule. Further, the agent measures the effectiveness of the selected method by monitoring reaction, and brushes up the profile.

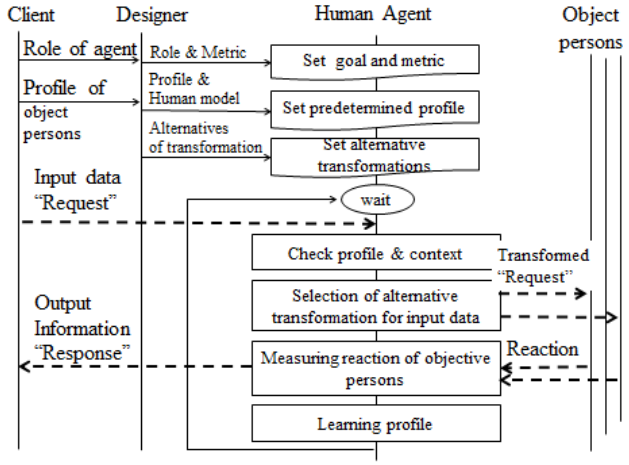



Figure 7. A design method of the human agent.
- Manual programming or Algorithm -

## VII. CONCLUSION AND FUTURE WORK

We have verified our hypothesis about human behavior through repeated experiments. One of the authors acted as the human agent to activate communication in an organization. As a result, we have found the effective messaging style that got about 1.5 times of responses by understanding predetermined profiles such as individual name and email address, and organization structure. Next, we have categorized examinees' profiles into "stubborn ones" and "flexible ones". The behavior of examinees with stubborn profile could not be changed by a small number of messages. However, the examinees with flexible profile became reactive by requesting it at their available time. So, we conclude that human communication can be activated by figuring out the profile and the context of the object persons. Therefore, the human agent should have the knowledge about the context as well as the profile of the object persons. Further, we have clarified that the agent was able to brush up their profile by observing the actual reaction of the object persons. Finally, we classified the procedures in our experiments into manually coded by designers and automatically implemented by algorithm implemented in the agent to address our design method of the human agent.

In the near future, we will refine the suitable location of processing entity to execute the agent program. In this paper, we assumed a stand-alone agent between the client and the object persons. Considering that a huge number of agents play various roles in the entire network in the future, the communication between agents will be required. This is also an open issue.

## ACKNOWLEDGMENT

We appreciate all the volunteers in Fujitsu Laboratories for joining our experiments. Further, we would like to thank Mr. Makoto Murakami, Mr. Akio John Iwata, and Mr. Teruhisa Ninomiya for their encouragements and helpful suggestions.

## REFERENCES

[1] NETCRAFT "Web Server Survey," January 2017. https://news.netcraft.com/archives/2017/01/12/january-2017-web-server-survey.html , retrieved April, 2019.

[2] Internet 2012 in numbers. https://royal.pingdom.com/internet-2012-in-numbers/, retrieved April, 2019.

[3] IEEE communication society "COMSOC 2020 Report," pp. 59-62, December 2011.

[4] H. Yamaoka, et al., "Dracena: A Real-Time IoT Service Platform Based on Flexible Composition of Data Streams," 2019 IEEE/SICE International Symposium on System Integrations (SII 2019).

[5] S. Rajeev, "Seven Sense Technplogy," 2015 IEEE UP Section Conference on Electrical Computer and Electronics.

[6] C. E. Shannon and W. Weaber, "The Mathematical Theory of Communication," University of Illinois Press, Urbana, 1964.

[7] S. K. Card and T. P. Moran, "The Psychology of Human-Computer Interaction," Allen Newell, 1983.

[8] M. Katoh, et al., "Proposal of a Personal Agent for Human Centric Information Networking," ICOIN 2018.

[9] M. Wiser "The Computer for the 21st Century" Mobile Computing and Communications Review, Volume 3, Number 3, 1991.

[10] M. Katoh, A. Okada, and T. Kato, "The concept and model of 4 dimensional traffic engineering," ICNS 2006.

[11] M. Katoh, Y. Tajima, H. Senoo, and D. Kimura, "Context aware cell selection in heterogeneous radio access environment," AINA2017.

[12] I. Iida and T. Morita, "Overview of Human-Centric Computing," FUJITSU Sci. Tech. J., Vol.48, No2, pp.124-128, April, 2012.

[13] S. Kurihara "Artificial Intelligence" IEICE 100th Annalistic publication Section 2, 6.3 p388-389, (in Japanese) September, 2017.

[14] T. Nishigaya, T. Kurita, I. Iida, and K. Murakami, "Proposal of Agent-based Network Architecture," IEICE Trans., B-I Vol. J79-B-I No.5, pp.216-225, 1996.

[15] T-C Huang, C-S Yang, S-W Bai and S-H Wang, "An Agent and Profile Management System for Mobile Users and Service Providers," AINA 2003.

[16] A. Yassine, A. A. N. Shirehjini, S. Shirmohammadi, and T. T. Tran, "An Intelligent Agent-Based Model for Future Personal Information Markets," 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2010.

[17] S. Ismail, M. S. Ahmad and Z. Hassan, "Regression Analysis on Agent Roles in Personal Knowledge Management Processes -Significance of a Connect Agent in Mediating Human's Personal Knowledge Management-," CITA2013.

 ISBN: 978-1-61208-711-5 

[18] S. Oishi, and N. Fukuta, “Toward a Flexible Ability Selection Mechanism for Personal Assistant Agent using Ontology Reasoning,” 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops, 2016.

[19] K. Urakawa and T. Sugawara “Reorganization of Agent Networks with Reinforcement Learning based on Communication Delay” 2012 IEEE/WIC/ACM, International Conference on Web Intelligence Workshops, 2012

[20] https://www.nttdocomo.co.jp/service/mydaiz/function/ (in Japanese) retrieved April, 2019.

[21] J. Hill, W. R. Ford and I. G. Farreras, “Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations, ” Computers in Human Behavior 49 (2015):pp 245-250.

[22] H. Ando “Sensibility Information Processing” IEICE 100th Annalistic publication Section 2, 6.1 p381-384 , (in Jpanese) September, 2017.

[23] M. Katoh, M. Murakami, A. Yamada, J. Suga, M. Kawai and M. Murakami “Views about Context Networking (3)” IEICE Technical Report NS2018-8, (in Japanese) April, 2018.

[24] Japan Business Association, “Status survey of e-mail for busines, 2016” (July 1, 2016) (in Japanese) retrieved April, 2019. http://businessmail.or.jp/archives/2016/07/01/5668

## APPENDIX

Table A-I and Table A-II shows actual contents of the opinion survey, i.e., question and choices for answer for our experiments I and II, respectively.

TABLE A-I. INQUIRIES AND CHOICES FOR ANSWER FOR EXPERIMENT I

| Trial | Inquiry | Choices for answer |
|---|---|---|
| #1) 1/24 | Memorized telephone number | a) 0~2, b)3~5, c) more than 5 |
| #2) 1/29 | Communication tool for close friend or family | a) Phone, b) Chat, c) Others including e-mail |
| #3) 2/1 | Number of transmitted e-mails per day | a) 0~10, b) 11~20, c) more than 20 |
| #4) 2/6 | How many terminals can you have for a walk? | a) 0~1, b) 2, c) more than 2 |
| #5) 2/14 | Can you allow AI to join your meeting? | a) Yes, b) No, c) Others（case by case） |
| #6) 2/20 | What impression if AI compliments you? | a) Good, b) Not good, c) Others（timing） |
| #7) 2/28 | Have you heard the terminology “network effect” and “information asymmetry”? | a) None, b) One, c) Both |

TABLE A-II. INQUIRIES AND CHOICES FOR ANSWER FOR EXPERIMENT II

| Trial | Inquiry | Choices for answer |
|---|---|---|
| #1) 5/24 | Do you like to listen to music during your job? | a) Often, b)When it is noizy, c) No |
| #2) 5/29 | Where do you have lunch? | a)In office, b)Restaurant in company, c)Restauramt outside, d)Others |
| #3) 6/4 | How to get news? | a)Broadcast, b) Internet, c) Newspaper, d) Others |
| #4) 6/14 | Expression that makes you feel respected | a) Thanks, b) Interesting, c) Admire, d) Others |
| #5) 6/21 | Expression that makes you feel discouraged | a) No response, b) Not interesting, c) Consecutive questions, d) Others |
| #6) 7/17 | Are you satisfied with communication? | a) Enough, b) Not enough, c) Others |
| #7) 8/1 | What do you feel happy the most during R&D activity? | a)Achievement, b)Discovery, c)Acceptance, d)Others |
| #8) 8/7 | What is an obstacle to your R&D time? | a)Regulation, b)Side job, c)Private issue, d)Others |

# Using Big Packet Protocol Framework to Support Low Latency based Large Scale Networks

Kiran Makhijani, Renwei Li, Hesham El Boukary

Future Networks, America Research Center
Huawei Technologies Inc., Santa Clara, CA, USA
email: {kiran.makhijani, renwei.li, hesham.elboukary}@huawei.com

***Abstract*—The performance of many automation-based services over networks continue to demand lower latency and higher reliability. With Industry 4.0 initiative, the scale of such applications will grow over time requiring large-scale *High Precision Communications* as its foundation. IP-based networks with existing service delivery models do not support time related guarantees. In contrast, Time Sensitive Networking (TSN), which is a data link layer technology, supports many paradigms of *High Precision Communications* but capabilities are limited to subnets with only a few number of connected devices. A network layer (IP-based) solution is needed to overcome limitations of flooding and control overheads in layer 2, in order to expand the use of TSN services over broader domains. In particular, an extensible IP-based data plane approach exploiting already available hardware capabilities of TSN solution can be envisioned. This paper discusses one such approach using Big Packet Protocol (BPP) and develops a cross-layer forwarder method to combine benefits of BPP high-precision directives in network-layer with time-sensitive capabilities of TSN.**

***Index Terms*—Big Packet Protocol; BPP; High-Precision Networking; Programmable Networks; SLA.**

## I. INTRODUCTION

Low latency applications are fast gaining mainstream momentum at large scale. That is, they are not limited to a single factory floor or studio networks, purpose-built for use with proprietary services. The number of such applications is continuously growing not only in private network domains such as factories, but are expanding to mass-consumption in service providers' networks as well. Interfaces such as human-to-machine and machine-to-machine are the basis of next generation connected services that aim to deliver digital world interactions with a very real user experience. Several such applications are sensitive to the precise time of delivery of information.

In networks, a layer-2 mechanism is offered by TSN protocol suite in bridged networks and has seen wide adoption in industry control, automotive networks and Audio Video Bridging (AVB). However, this can easily become difficult to scale as the density of end-stations, and bridges go beyond a certain number. A further proliferation of several latency-bound applications in service provider infrastructures is expected with the onset of 5G network slices for verticals such as Vehicle to Everything (V2X), critical infrastructure, Internet of Things (IoT) and so on. Many such service verticals need to stretch beyond layer-2 boundaries in order to scale better. Therefore, a TSN-like support in IP-based networks will be necessary.

High precision communications are the ones that guarantee packets of a flow associated with a service are delivered accurately on or before the prescribed time. Networks that are capable of providing high-precision communications are expected to have necessary network resources on each node in terms of buffers and deploy deterministic scheduling algorithms to achieve time-guarantees of such services. However, for the most part, packet forwarding technologies in large-scale IP-based and/or traffic-engineered networks continue to serve only statistical resource requirements, i.e., allocating from shared resources. This often comes with high cost of provisioning of network elements and their resources. To operate at large scale for diverse set of applications over larger area, minimizing configurations while providing higher level of customization and finer granularity of resource specification is necessary. It requires additional capabilities to be defined for IP, which is done via BPP.

In this paper, we propose a generic routed network solution based on BPP and extend the use of existing low latency TSN bridged networks. This paper describes the use of BPP framework [1] (a.k.a. New IP) to provide high-precision communications specifically for bounded-latency applications as covered under TSN. The New Internet Protocol (New IP) or BPP delivers high precision services over IP networks. We explore BPP as a solution to deliver time-sensitive services in layer-3 domains.

The New IP (will be referred to as BPP in remainder of the paper) defines high-precision communications suite which comprises of (a) in-time, (b) on-time, and (c) coordinated delivery of services - all of which are factor of time. It is a network layer solution that may easily be deployed at scale by any application. BPP is a new technology that provides building blocks both for customizing data plane forwarding from a user's perspective as well as in-node mechanisms to process many network parameters to manage packet latency and scheduling. At the same time, TSN is a well-established ethernet-based protocol suite built on the foundations of real-time Ethernet, e.g. Profinet, EtherCAT, etc. TSN is a part of IEEE802.1 standard and is widely used in AVB studio and factory floors networks. It consists of well-designed resource reservations and scheduling algorithms to support end-to-

 ISBN: 978-1-61208-711-5 

end bounded latencies. We demonstrate how TSN can be expanded to provide Ethernet services at a higher layer, while simplifying operation, control and monitoring.

This paper makes the following contributions: (1) introduces fundamental requirements for high-precision services with respect to the growing demand for latency-sensitive applications over bigger regions, (2) provides an overview of capabilities of TSN protocols along with their limitations, (3) provides a vision of BPP router node to support high-precision forwarding paradigm, and (4) finally elaborates a cross layer forwarder to combine capabilities of TSN with BPP to extend them to wide area applications.

The paper is organized as follows: Section II describes the motivation behind our work and provides a background to time-sensitive networking. Section III is an in-depth discussion of high-precision services and discusses BPP technology as means to achieve such services. Section IV very briefly discusses related work. Sections V and VI discuss in detail the contributions of this paper, finally covering future work in this area under Section VII.

## II. Background And Motivation

In industry operations, typical requirements for automated control of production floor requires bandwidth ranges of the order 100M-1G with latency 1ms to 200ms (it may even be lower for isochronous control such as PLC and embedded control) requiring interactions with Cyber Physical Systems (CPS) involving Machine to Machine Communications (MMC) [2]. Since these tight latency requirements originated from on-premise networks built on bus-like or LAN communications, the TSN solution was developed as a part of Ethernet protocol suite. Formally, IEEE 802.1 group defines TSN applications as those responding to external stimuli within a fixed, and often small, period of time.

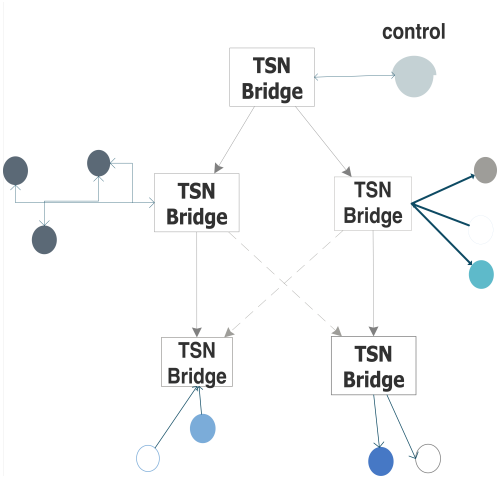



Fig. 1. Industry control network reference model

A factory floor will typically use a bridged topology as shown in Figure 1. The TSN-bridges connect with end stations and support usual layer-2 protocols such as spanning tree, VLAN etc. while providing latency guarantees in bridged network. TSN bridges provide bounded latencies, completely automated and reliable connections between the end stations such as machine equipment, sensors, PLCs etc. (shown as small circles directly connected with different TSN bridges Fig. 1 above) and a command controller shown as just another device in the network at the root bridge for simplicity but may be connected anyhere in this network. Such models have been widely deployed not only in industrial automation but also in Car Area Networks (CAN) and AVB production studios.

Contrary to this, Industrial 'internet' by very definition means interconnecting different end stations and industrial applications across multiple network domains, not just limited to local area networks. The principal goal of network slicing is to use common and public communication infrastructure for different types of services. An ultra-reliable low latency slice being just one 'type' of network-service and may have several instances based on market verticals. Obviously TSN by itself is not suitable for building applications at this scale. Either TSN should be extended beyond LANs or we need similar technology in layer-3 networks. Arguably, IP networks are best-effort and such bounded latency services were not inherently supported layer-3 until BPP [1]. BPP formally defines high-precision communication services as a group of technology and capabilities in the scope of layer-3 to serve similar but broader purpose than TSN. TSN is a well-known standard and provides several hardware capabilities for serving real-time applications but is not capable of scaling beyond bridged networks. Our contribution is to address this gap by introducing a cross-layer forwarder, which leverages information from both BPP (the network layer) and TSN (data link layer) to provide high-precision services in a generic manner. Our objective is to maximize re-use of existing TSN technology and build large scale high-precision networks with minimal disruption. We show that using cross-layer forwarder with minimal changes we can make use of many hardware components without much overhead. We then demonstrate the validate our approach by studying the forwarding path with BPP cross-layer forwarder.

## III. High-Precision Communication Services

Applications in automation, such as machine to machine interactions, vehicle to infrastructure, smart cities, remote surgical procedures, etc. have diverse and variable requirements from networks. While TSN primarily supports technology to guarantee worst-case end-to-end latency, there are in fact, more critical factors to time-sensitivity which are collectively described as High-Precision communication (HPC) services [3]. The HPC services can be broken down into a) in-time packet delivery - like TSN applications, b) on time service - with an extremely low delay variation between actual and planned packet arrival time and c) coordinated service - having more than one data stream arriving in specified bounds of time. This classification gives a better sense of criticality and time sensitivity in terms of how networks elements should treat such services. Regardless of the classification, there are certain

 ISBN: 978-1-61208-711-5 

common requirements to be met in delivering high-precision services and are described next.

### A. Requirements For High-precision Services

Firstly, a knowledge of different HPC profiles is needed based on which appropriate bandwidth and path computations can be done. There may be different service-operation profiles based on the resources required for each end device (bandwidth, latency, etc.). The same network is also integrated for Information Technology (IT) services, such as management, monitoring, telemetry collection, etc. which do not have stringent service delivery requirements and no special traffic treatment is necessary. However, they may contend for same resources in the network at random period of times. Therefore, a discrimination of HPC services from normal traffic is necessary.

The second requirement is that in industry automation all network nodes should have an ability to compute transmission scheduling criteria and position in queue with high accuracy on network nodes. It requires knowledge of both complete topology and the path associated with the streams, all of which need high-precision time synchronization in the network.

Third, flow classification is required for a network node to identify what kind of service profile the flow belongs to in order to process it as per the constraints of that profile.

Then, determination of timing behavior is necessary. Special time-aware flow processing functions are required to ensure all packets are forwarded with in the desired timed-accuracy. These functions reside in the network element and support different varieties of scheduling and shaping mechanisms suitable for both time-sensitive and normal service profiles. These functions operate with the knowledge to resource specification of a service profile and will include parameters, such as bandwidth, jitter, and latency.

Finally, reliability is of prime importance in industry control. If a packet is dropped in transit, entire synchronized and automated factory pipeline may come to a halt or even worse may lead to commands being processed out of order causing several anomalies in production. Therefore, at the network level, path redundancy and extreme reliability functions are very important.

### B. Challenges Beyond Switched Networks

TSN satisfies above requirements within the scope bridged-network (e.g. a few kilometers) for limited size of fixed topology through a suite of new provisioning and forwarding techniques, but there exist some limitations in delivering time-sensitive services.

**Scalability**. In traditional manufacturing, automation distances are few kilometers (less than 10) and limited number of devices, however as Industrial internet grows, it will be hard to scale the networks in layer 2 and even more complex to partition on per-service basis. The disadvantages of such large-scale bridged networks are well-known from data center networks (flooding, slow STP convergence, and so on), that have already transitioned to IP based solutions.

**Infrastructure sharing**. 5G brings automation in all kinds of applications and economy of scale demands that networks for edge services, V2X applications, critical services and industrial networks etc. shall share the infrastructure and hardware. Therefore, isolating applications is far simpler and scales better with IP based networks.

**Complexity**: A closer examination of TSN will show that it has evolved into a complex set of protocol suite. Several protocols have to be well understood and provisioned on all the bridges. The challenges of provisioning at scale, their response to topology changes, update and withdrawal of stream specific resources is an expensive procedure collectively generating several Bridged Packet Data Unit (BPDU)s and other control PDUs in the bridged network.

In comparison, BPP framework is a simple and customizable data plane for delivering high-precision services. The BPP has two primary artifacts, first that specifies what goes on the wire and signals per packet Service Level Objectives (SLO)s in a ***contract*** (BPP Block in Figure 2) to intermediate network elements, such as routers. Second, a programmable compute, forwarding and schedule engine on those routers to implement components necessary for scheduling and shaping functions. Because of these two factors, BPP is capable of combining on-the-wire contract with any hardware element that supports high-precision functions such as a TSN-bridge. We propose a simple method of cross-layer forwarder between BPP contracts and TSN scheduling and shaping functions by allowing coordination of technologies crossing IP and MAC layers. In the following section, we briefly describe end-to-end control and flow processing and forwarding based on TSN protocol suite.

## IV. RELATED WORK

Historically, IP services have been implemented using Diffserv [4], IntServ [5] and RSVP [6]. These techniques can provide bandwidth assurance but latency guarantees are not possible, especially with interfering traffic. The most common queuing discipline used in IP based networks is deficit round robin method. Traffic engineering mechanisms for IP focus mainly on providing paths based on certain resource requirements, mainly bandwidth guarantees but often low-latency paths may also be computed. Recently, Internet Engineering Task Force (IETF) formed a deterministic networking work group (DetNet WG) to address a similar problem of time-sensitive networking. The scope of DetNet [7] does not include pinning time-sensitive low-level capabilities on the nodes but covers only the data plane that carries the deterministic service. This is limiting because the aggregated resource reservations have to be made in advance for the DetNet flows. Finn [8] discusses computation of worst-case end-to-end bounded-latency, but refers to external queuing mechanisms without details on how to integrate them in IP-based networks.

While IP-based networks lack sophisticated queuing and scheduling capabilities, Ethernet based TSN scheduling algorithms are well-established and thoroughly defined. Thus, our approach to achieving high-precision communications, at

 ISBN: 978-1-61208-711-5 

large-scale on per-flow basis is based on utilizing capabilities of underlying time-sensitive bridges.

## V. High-precision Networking With BPP

### A. BPP Overview

BPP was first described in Big Packet Protocol framework [1]. The basic idea is that of injecting meta-information into packets to provide guidance to intermediate routers about processing those packets. This is done by attaching BPP Blocks (or contracts) with directives that provide guidance for the packet treatment such as what resources must be made available for the packet, as well as the flow that the packet is a part of. Rather than relying on in-built logic provisioned statically through management or control plane of networking devices that may result in best-effort treatment of the packet, a BPP network device will act on those directives and metadata to handle the packet, overriding any regular packet processing logic that is deployed on the device. This is in particularly important when dealing with resource-centric commands, for example, to determine conditions when to drop a packet, which queue to use, when to allocate a resource, or to measure a service level and compare it against its SLO.

This concept allows behavior of packets and flows to be programmed by injecting BPP Blocks (contracts) into packets at the edge of networks. There is no need to program networking devices or network controllers directly. At the same time, the programmed behavior is isolated from other flows and restricted to the packet and its flow. A BPP packet is

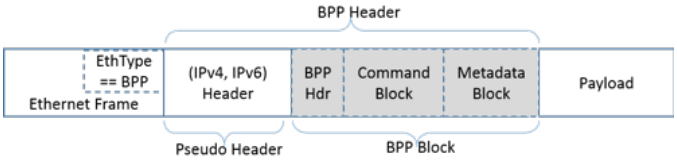



Fig. 2. BPP Packet Structure with Contract as BPP block

structured as depicted in Figure 2. It consists of a pseudo-header of its host protocol (pointing to BPP Block as next protocol), as well as one (or more) BPP Blocks.

### B. Sample Description Of High Precision Services

For example, the meta-information that BPP carries can easily be used to describe a high precision service requesting end-to-end latency of 14 ms and peak rate 512 kbps. In BPP contract, it requires two instructions 1) for latency-constraint and b) for peak rate as below. The actual encodings of these instructions are not described since they are implementation specific.

$$LatencyFn(intime|e2eFn(14, ms)|residual(time))$$
$$BandwidthFn(peakrate(512, kbps))$$

### C. BPP Forward Processing On The Node

BPP contract processing requires a forwarding component that is capable of parsing and understanding the semantics of the contract carried by BPP. A simplified BPP node is shown below in Figure 3. In this figure, an enhanced BPP forwarding plane is shown with all the usual forwarding plane tables and components such as packet memory, classification tables, forwarding information base (FIB), Access Control List (ACL), and policy tables along with port specific queue manager, scheduler and shapers. In addition, there is a BPP parser that parses the instructions, metadata and state in the BPP Block, generating an output result that can be directly used by queue manager to schedule packets in output queue. Essentially, we show clear separation of three blocks: (a) BPP parser and compute, (b) Traffic manager, (c) lookup and state tables.

As a packet is received on a port, it is classified and checked for any ingress filtering, then BPP contract processing of different instructions happens based on the outcome of parsing logic. The results of instructions are generated and fed into the final scheduling and queuing functions. This forwarding pipeline is quite similar to Ethernet forwarding Section V-E described later, just that the functions are specific to layer 3 headers and in Ethernet, they are layer 2 port specific.

### D. Leveraging Time Sensitive Networks Capabilities

A comprehensive detail of IEEE 802.1 TSN Task Group (TSNTG) [9] work is presented in [10] survey and therefore, we do not discuss details of the individual protocols and only cover the broader area in the context of the paper. A summary of TN work is shown in Table I.

**Provisioning and reservation**: The reservation of resources is mandatory in TSN networks. To this end, the reservation protocols have been enhanced to support centralized (Centralized Network configuration (CNC)), distributed (Stream Reservation Protocol (SRP)), and hybrid modes. SRP utilizes signaling between talkers, stations producing streams and advertise network resource attributes of the stream towards listeners, the devices consuming those same streams and declaring resources available for their reception. Reservations are created when these events are combined in a bridge.

**Classification and marking**: There is a new SR traffic class associated with two additional queues that enjoy higher precedence than usual priorities or priority code points. THe default settings map priority 2 and 3 to SR class B and A, respectively. In distributed SRP, bridges use credit-based shaper (CBS) data plane (Table I second row). The end stations are required to mark the packets with SR class A or B. Later, stream configuration enhancements were introduced

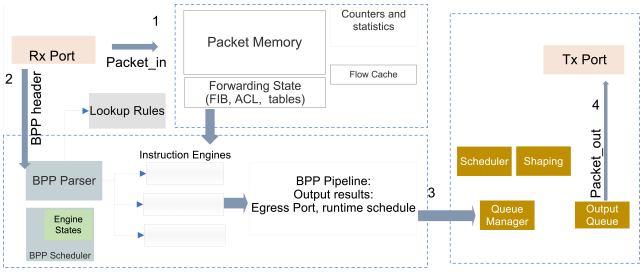



Fig. 3. New IP functions on Network Element

to support all TSN functions such as shaping, preemption, and redundancy through the CNC model. RAP is introduced to do distributed reservation for hard real-time control type applications. LRP provides local port services.

**Packet scheduling and shaping**: As shown in Table I first row, TSN supports different types of scheduling and shaping algorithms to address different functional requirements for bounded latency with rate constrained traffic, scheduled and best-effort traffic. Both queue and transmission algorithm selections are done based on traffic class. Credit-based shapers handle near real-time traffic which do not satisfy industry control requirements of tighter and lower latency guarantees. These are requirements are met using time-aware shapers with repeating schedules at pre-determined intervals. In-traffic-class interference among time-critical streams has to be further eliminated.

**Synchronization and Reliability:** Ethernet systems use 802.1AS Timing and Synchronization for Timing-Sensitive Applications, and 802.1CB Frame Replication and Elimination for reliability.

### *E. TSN Forward Processing On The Node*

End to end forwarding of time-sensitive packets is well integrated in 802.1Q bridged networks. The MAC bridges are associated with separate learning, filtering and forwarding functions on receiving ports along with egress filtering, transmission selection and queue management functions taking place on transmit port as seen in Figure 4. The information required for forwarding comes from different dynamic and management configured tables. For example, implementation of the decisions governing where each frame is to be forwarded is determined by the relay function using forwarding rules that are populated in the Filtering database (FDB). The operation of relay function includes verification of active topology, classifying frames to expedite time-critical traffic, and frame format conversion for destination stations.

### *F. Layer-2 Service to Cross layer forwarder*

Our motivation is to reuse queues of TSN ports with forwarding logic of BPP to provide scalability to time sensitive

TABLE I. AN OVERVIEW OF DIFFERENT TRAFFIC CLASSES IN TSN

| **Traffic class** | **Best Effort** | **Rate-constrained** | **Scheduled** |
|---|---|---|---|
| *Data plane techniques* | Strict priority algorithm | Traffic shaping with Credit-based Shaper, 802.Qav | Time-aware scheduler, 802.1Qbv-Scheduled traffic, 802.1Qch Cyclic Queuing, 802.1Qcr Async Shaping |
| *Control plane techniques* | STP & so on | Dist. stream config bandwidth reservation resource allocation using Stream Reservation Protocol (802.1Qat) | Centralized stream config path, schedule calculation. management with a central controller (802.1Qcc) |
| *Target Latency* | Non-deterministic | Bounded max. latency | Guaranteed lowest latency |

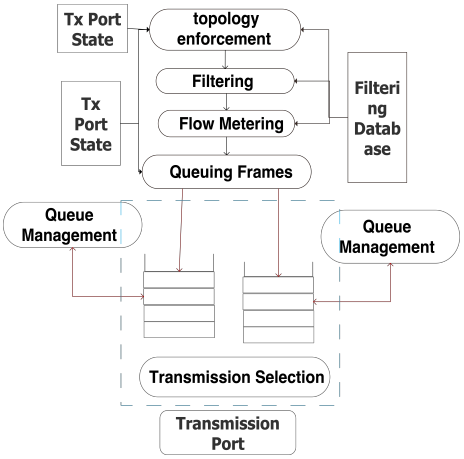



Fig. 4. TSN Forwarding process

applications by using IP networks. We see benefits with the use of pinned-down hardware capabilities of TSN, and do away with layer-2 protocols, forwarding rules, policies and filter tables in favor of higher resolution of service control in layer-3. Ethernet bridges support relay function in which packets from a receive port are processed and sent to the transmit port. The bridge design provides information to relay function or tables in 3 ways, 1) learnt dynamically through incoming port, 2) management interface, 3) higher-layer MAC Service to higher layer entities using bridge port functions. In all the above scenarios, the user of relay functions essentially provisions various layer-2 tables for relay function with different forwarding, filtering, and policing rules.

None of the above three options are usable with BPP, nor do they provide the value of scale and simplicity we aim to achieve. Therefore, our proposal is to connect bridged ports with BPP forwarding pipeline, as explained in next section.

## VI. EXTENDING TSN WITH NEW IP FORWARDER

In addition to looking at TSN bridges, we investigated well-known classical methods of cross-layering; thematically, in these approaches each layer continues to perform its function, while those functions are improved or optimized for the purpose of overall application response time or quality of experience. We found that cross-layering research is more significant either at transport level [11] [12] for better integration with applications, or in wireless networks [13] [14] for better feedback about signal strength and availability between MAC and PHY.

We did not come across a lot of literature on cross-layering between layer-2 and layer-3. There may be several reasons behind this such as ossification, specific segregation of switching and routing domains (i.e. either switched or routing policies, rules and management is used) etc. However, we believe the main reason is that in deployed communication protocol stack, over time layer-2 and layer-3 forwarding and control methods have essentially evolved independently. Yet, interestingly, the foundations of forwarding, policing, and scheduling are often found to be quite similar. For example,

 ISBN: 978-1-61208-711-5 

commonalities are seen in 'forward to next hop' functions; both layers use destination address based lookup, priorities are marked in packets header, and in particular development of TSN solution, several IP protocols were used as a reference to design similar requirement in TSN protocols such as, RSVP & SRP, and PTP & Time synchronization. Thus, in building high-precision services with BPP, we find the greatest benefit in cross-layering of 'functions' instead of information. Doing so, we combine the good of layer-2 and layer-3 as:

- Forwarding functions of New IP, since they provide high-degree of customization, replace forwarding of TSN
- Scheduling functions of TSN, since they exist already in deployments are reused.

As shown at the top part of Figure 5, functions of BPP parsing and forwarding are linked with the queue management functions of TSN in the same figure. We do not show different tables that will be populated and managed at runtime because BPP is the runtime execution-pipeline; it is not a chain of different tables in the traditional sense of forwarding pipelines.

We use directives from BPP block, such as those defined earlier in section V-B, to process high-precision requirements, run them through the BPP parser and processing engine, derive the result as to which algorithm (among the TSN supported algorithms) guarantees service, determine queue, and its parameters to schedule the packet in TSN queue component. BPP parser and processing engine determines the algorithm, the egress port, and the schedule. The BPP traffic manager entity maps schedule to traffic class and interacts with TSN queue management function.

### A. Cross-layer forwarder Initialization

***TSN Queue Manager/Schedule configuration***. The way queue management (QM) algorithm will work is decided at the management plane. i.e., the operator determines what kind of scheduling behavior is required and accordingly TSN QM can be configured. TSN provides several managed objects and external configuration parameters through which queue, traffic class and algorithm specific information can be programmed in the TSN switch. Through configuration knobs, the exact timing behavior of queues can be provisioned such as bandwidth allocation for a particular traffic class, gate control list, queue max Service Data Unit (SDU) and even the list of algorithms supported. Queue configurations are completely unrelated to forwarding tables such as FDB and learning tables.

***BPP Traffic Manager***. Initialization for BPP is only needed to determine mappings in traffic manager, the traffic classes are to be mapped to the ones configured by TSN QM; BPP does not require any markings like .1P priority bits or even VLAN tagging. The result of the parser can determine the traffic class as an internal parameter (similar to internal priority value in TSN). ***Resource Reservation***. No TSN or IP reservation algorithms are used for profiles of different type of services. This is because the Tspec equivalent information is carried along with the packet in New IP (in-band). The hint for reservations happens in flow cache of BPP engine with first packet. The BPP schedules to send the packet according to embedded TSpec not according to what is reserved on the node. This is possible because BPP traffic manager uses cumulative port state based on queue-depths, if a particular packet can be sent in time.

### B. Runtime Forwarding path

Figure 6 below shows a high-level packet processing and forwarding through BPP block to determine how latency instruction can be mapped to a queue or a traffic class and what algorithm can be used. As a high-precision service packet is received, the following processing is done.

1) In a high-precision network (large- or small-scale), an operator has a knowledge of the type of service. At the head-end node when packet arrives, a BPP block is injected with the high-precision service profile. BPP-Header insertion is a function of forwarding pipeline installed as a police on edge routers (not shown in the 6).
2) For an incoming packet, layer-3 lookup
   - uses FIB to determine the egress port

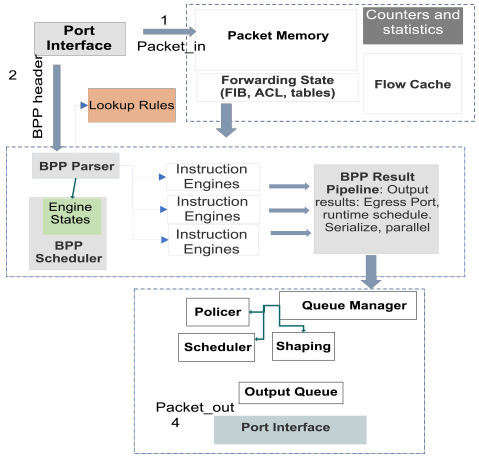



Fig. 5. Cross layer New IP forwarder functions

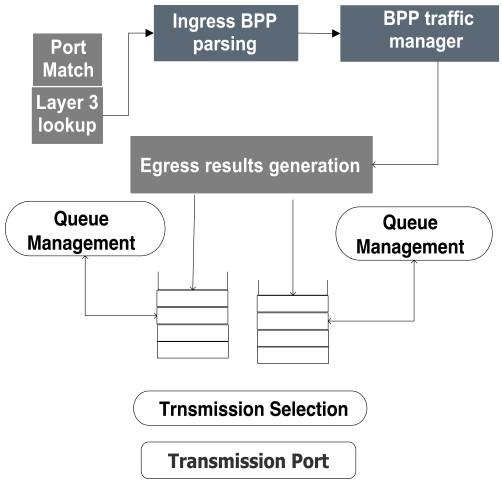



Fig. 6. Cross layer forwarder: packet processing

- recognizes that it is a BPP packet and dispatches it to BPP parsing engine. BPP parsing engine sees latency directive, determines type of high-precision service, and sends to egress results generation.
- Egress results generation combines FIB and BPP results to assign appropriate algorithm and determines traffic class.

3) Internal state of BPP engine maintains the knowledge of resource budgets and keeps 'available resource' repository updated.
4) BPP traffic manager function finds the mapping to traffic class, peeks into queue tables to find runtime queue depth for that traffic class and determines if the latency-goal is feasible or not; if yes, it transmits the packet to that queue.
5) On the tail-end before delivering the packet to end-station, the BPP block is removed and plain IP packet is delivered.

### C. Solution Analysis

The cross-forwarder solution is examined briefly in the following manner:

*Shorter lead times:* BPP is a new technology in which many new function components need to be developed. BPP framework requires hardware-support to implement high-precision networking and TSN switches can fulfill this gap. Given the cost and lead time for procuring prototypes, BPP benefits from the reuse of existing well-proven hardware schedulers.

*Reuse Benefits*: We proposed a theoretical model to tweak existing layer-2 forwarding pipeline in the layer-3 pipeline and make it usable with IP-based networks. The forwarding pipelines are based on match table look ups and assigning proper egress queues. Such tables can be programmed from the control plane with vendor-specified interface or drivers without having the need to change scheduling behavior or spin a new hardware. This was demonstrated via Figure 4 and Figure 6. It allows for new services to describe their service requirements with a fine granularity using BPP and have them treated using time aware schedulers.

*Forwarding path validity*: We provided a systematic examination of the forwarding path and corresponding BPP components necessary to develop the cross-forwarder. It exploits the fact that queue identifiers in the TSN switches are independent of link layer specific addressing. The feasibility of the cross-forwarder can be proven through empirical analysis which will be our next area of focus.

*Performance:* The performance comparison between TSN switch and proposed cross-forwarder will depend on the cost of processing BPP directives versus BPDU processing and is a part of our future work.

## VII. FUTURE WORK

So far, we have presented the feasibility concept of cross-layer forwarder. Although this discussion is quite thorough, it still needs validation with implementation. There are not great options of opensource TSN algorithms or comprehensive SDK for the existing TSN switches. At the time of writing this document, New IP development work is in progress; once available it can be used to integrate and further evaluate our approach presented in this paper. While it is simple to develop this concept in software, it is still necessary to explore the amount of driver or FPGA changes required to use TSN switches.

## VIII. CONCLUSION

As is evident from the previous section that TSN solution requires several protocols leading to overall higher operational complexity. The biggest limitation remains that it is only a layer-2 solution; in order to scale over wide area networks, a network layer approach is desired. In this paper, we propose that new data planes like BPP can tremendously reduce provisioning protocol complexities. We demonstrated emulating a TSN switch as a layer 3 high precision router is feasible which will allow a fast adoption of high-precision services in networks.

## ACKNOWLEDGEMENT

The authors would like to thank members of their team who provided valuable information. These are Lin Han, Lijun Dong, Padma Esnault, Yingzhen Qu, Uma Chunduri, Toerless Eckart, and Alexander Clemm.

## REFERENCES

[1] R. Li, A. Clemm, U. Chunduri, L. Dong, and K. Makhijani, "A new framework and protocol for future networking applications," *ACM Sigcomm Workshop on Networking for Emerging Applications and Technologies (NEAT 2018)*, pp. 637–648, May 2018.

[2] M. J. Teener, "Keynote: A time-sensitive networking primer: Putting it all together," *International IEEE Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS*, 2015.

[3] R. Li, "Network 2030: Market Drivers and Prospects." "https://www.itu.int/en/ITU-T/Workshops-and-Seminars/201810/Documents/Richard_Li_Presentation.pdf", Oct. 2018. First Workshop on Network 2030, New York University, Brooklyn, US.

[4] D. Black, and P. Jones. Ed., "Differentiated Services (Diffserv) and Real-Time Communication," *IETF, RFC 2205*, Nov. 2015.

[5] Y. Bernet et al., "A Framework for Integrated Services Operation over Diffserv Networks," Nov. 2000.

[6] R. Braden et al., "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," Sept. 1997.

[7] N. Finn, P. Thubert, B. Varga, and J. Farkas, "Deterministic networking architecture," *draft-ietf-detnet-architecture-09 (work in progress)*, September 2018.

[8] N. Finn, E. Mohammadpour, J. Le Boudec, J. Zhang, B. Varga, and J. Farkas, "Detnet bounded latency," *draft-finn-detnet-bounded-latency-02 (work in progress)*, October 2018.

[9] "Time-Sensitive Networking Task Group." http://www.ieee802.org/1/pages/tsn.html. Last accessed 14 July 2018.

[10] A. Nasrallah et al., "Ultra-low latency (ULL) networks: A comprehensive survey covering the IEEE TSN standard and related ULL research," *CoRR*, vol. abs/1803.07673, 2018.

[11] S. Denny et al, "QoE Analysis of DASH Cross-Layer Dependencies by Extensive Network Emulation," *Internet-QoE '16: Proceedings of the 2016 workshop on QoE-based Analysis and Management of Data Communication Networks*, pp. 25–30, Aug. 2016.

[12] X. Corbillon et al, "Cross-layer scheduler for video streaming over MPTCP," *Proceedings of the 7th International Conference on Multimedia Systems (MMSys '16), ACM, New York, NY, USA*, pp. 1–12, Aug. 2016.

 ISBN: 978-1-61208-711-5 

[13] S. Ket, V. Shinde, R. Khandare, and R. N. Awale, "Cross layer communication for wireless networks," *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3 '09). ACM, New York, NY, USA*, pp. 629–632, 2009.

[14] F. Foukalas, V. Gazis, and N. Alonistioti, "Cross-layer design proposals for wireless mobile networks: a survey and taxonomy. Commun. Surveys Tuts," *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3 '09). ACM, New York, NY, USA, 629-632*, p. 70, Jan. 2008.

 ISBN: 978-1-61208-711-5 

# Software Defined Networking Managed Hybrid IoT as a Service

Peter Edge
Department of Computing
Ara Institute of Canterbury
Christchurch, New Zealand
email: peter.edge@ara.ac.nz

Zara Davar (Zahra)
Department of Teaching, Learning and Design
Ara Institute of Canterbury
Christchurch, New Zealand
email: zara.davar@ara.ac.nz

Zhongwei Zhang
School of Computational and Environmental Sciences
University of Southern Queensland
Queensland, Australia
email:zhongwei.zhang@usq.edu.au

***Abstract*— In the new era, communication devices use the Internet and World Wide Web to communicate from different locations around the world. The Internet of Things (IoT) extends this communication paradigm within different smart devices by collaborating sensor technology. In this model, infrastructure components must manage the large amounts of data generated by the smart devices and sensors. Integration of cloud computing with the IoT has many benefits and challenges; for example, cloud computing can improve the management of data from the collection phase to data process and backup. The most prominent challenges resulting from the integration are privacy and security. In this paper, we propose a secure hybrid cloud architecture mix with edge and fog computing to address security and privacy issues of IoT data. Our approach is to distinguish public and private data in the device data collection layer and address them to the right cloud (public or private) taking advantage of Software Defined Networking (SDN) for design and management of the networking layer. The privacy and security issues will be addressed within the design of the networking layer, in which all the necessary rules and protocols are in place and implemented.**

***Keywords-Internet of Things; Hybrid Cloud; Security; Privacy and Software Defined Networking.***

## I. INTRODUCTION

The new era of digitalization and communication aims to connect smart devices and real objects via the Internet. The landscape of Internet-based communications has been dramatically changed by the IoT [7]. IoT relies on intelligent devices interconnected within a dynamic global network infrastructure using the sensor technology to communicate with other smart devices [1] [8]. It is possible to use the IoT technology to create "robots" out of devices surrounding us, to collect data from the smart devices, and then to make intelligent decisions in our day-to-day life [2].

IoT mainly uses cloud computing for data collection and management. Even though cloud computing and IoT are two different technologies, they have a complementary relationship in collecting and processing a huge amount of data.

On the one hand, the cloud is predominately the platform utilised for Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [9]. Also, most known cloud types are public, private and the hybrid clouds. The hybrid cloud is a mixture of a public and a private cloud. On the other hand, cloud computing involves the on-demand delivery of computer power, database storage, applications and other compute resources.

IoT requires the flexibility of resource design in its architecture. The resource design must cover large scale storage for massive amounts of IoT data generated by devices [7], although IoT uses cloud computing architecture to solve many of the IoT computational and resource issues. However, integrating cloud and IoT technologies presents challenges, such as scalability, identification of different type of data, and the management of unnecessary data, heterogeneous networks, security and privacy.

This paper provides an overview of existing cloud solutions for IoT security challenges as well as our proposed solution. The remainder of the paper is organised as follows: Section II includes basic concepts; Section III discusses existing solutions; our proposed solution, the integration of the software definitions of networking and hybrid IoT is presented in Section IV; Section VII concludes the paper.

## II. BASIC CONCEPTS

In this section, we introduce the fundamentals of cloud computing and Software Defined Networks (SDN). Cloud computing refers to a network of remote servers hosted on the Internet [3]. It has been divided into two categories: public and private cloud.

In the infrastructure design for the private cloud, single tenant physical servers are often the best choice. In this paper, we call single tenant physical servers bare-metal servers. Bare-metal servers are dedicated servers assigned to each client without any resource sharing. One of the main benefits of using bare-metal servers, besides performance, is security. A bare-metal server physically isolates your data, applications and other resources [6]. Using bare-metal servers will help in achieving high performance and a secure environment.

On the other hand, the public cloud uses virtual servers. In this model, computing and storage are shared by different users. This will decrease security and privacy as well as performance. One of the main benefits of using a public cloud is having a cost-efficient cloud environment. Hybrid cloud is a term used in cloud computing and refers to a cloud architecture consisting of both the public and private clouds. SDN, simply defined, is the physical decoupling of control

 ISBN: 978-1-61208-711-5 

and data planes within traditional networking elements. While the control plane is responsible for routing path decisions, the data, or forwarding plane, forwards packets based on the logical knowledge of the control plane.

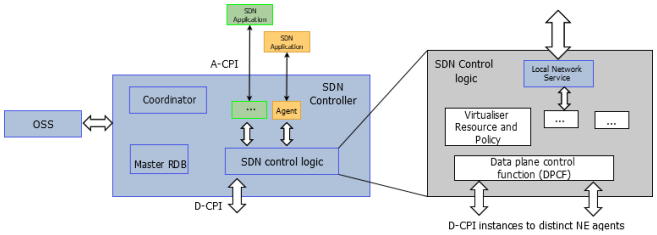



Figure 1. SDN controller logic

The result is a distributed model comprising a single controller influencing multiple forwarding devices. A representation of decoupling control and data planes is given in Figure 1. The biggest advantage for separating planes being the ability to control match criteria. Forwarding rules and flow match information can be injected via Application Programming Interfaces (APIs) available on the controller and distributed to forwarding devices via a secure link between controller and forwarder. OpenFlow is one such protocol utilised between controllers and switches.

Hybrid IoT as a service leverages the agility of what an OpenFlow-enabled network can offer. The ability to control flows is based on existing and extensible match fields. The programmability simplifies traffic engineering, providing an opportunity to craft custom match combinations and include priorities and action lists, or the ability to punt a matching packet to secondary match fields and actions.

## III. EXISTING SOLUTIONS

In this section, we present a critical review on the existing cloud computing solutions and discuss the gaps in the data challenges and security.

In an open source architecture called “OpenIoT”, web servers use sensors for data communication with the cloud [10]. In this solution, sensor communication with the cloud is through Representational State Transfer (REST) services and Simple Object Access Protocol (SOAP) protocol. This solution is based on the public cloud, which cannot cover security aspects for IoT data.

The Secure, Hybrid, Cloud Enabled Architecture for IoT (SHCEI) [11] solution presents a secure hybrid cloud design for IoT data security. In this architecture, the pure private cloud is placed in the device layer to collect IoT data. This solution provides a highly secure cloud solution for the IoT. However, using this approach generates an overload of unnecessary data. This will increase the number of cloud resources needed to manage the IoT data; this solution is not cost effective.

The idea of IoT data monitoring in an SDN-coordinated IoT-cloud has been introduced to ease the issue of data congestion in the IoT model [15]. In this research, using SDN flow steering makes available multiple paths for message delivery in IoT data usage, and performs monitoring of the data path in the network transport layer by using open source technologies. The problem with the design is how to distinguish and encrypt private IoT data before monitoring and delivering.

A solution proposed recently to address IoT data traffic is known as edge IoT analytics [16]. This research used SDN to manage data analytics at the edge cloud, stopping unnecessary data transfer to the next layer.

Meanwhile, another similar study proposed an IoT-aware SDN solution [17] to solve IoT data traffic congestion in the network edge. Even though these studies tackled the issue of IoT data traffic during transfer, synchronization between cloud components, SDN and IoT devices either had not been considered or is not an optimal and practical approach.

In another study, a Tenant Network (TN) has been proposed provide security in a multi-tenancy cloud environment for IoT data [18]. The idea of isolating all the network components to different zones such as the cloud controller, cloud administrator and tenant administrator was presented. This research improves trust between the cloud user and provider using TN architecture, although it cannot support distributed deployment with different controllers. Therefore, the approach cannot satisfy the IoT data scale.

In Edge Computing (EC), allocated applications, hosting happens at the edge servers [12]. EC is compatible with “private devices” such as smart phones, laptops, pagers, etc [13]. The aim of EC is to create a better quality of service for end users [13]. On the other hand, Fog Computing (FC) processes data at the LAN [14]. Therefore, fast and reliable data communication happens in FC. Both EC and FC will be used as part of our proposed architecture. Even though they are beneficial for IoT data collection and process, they need smart network architecture for the IoT data scalability issue. We will discuss this in Section V.

Although using cloud technology eases the management of IoT in many ways, there remain open gaps and challenges in this domain. Challenges such as data/resource management, communication, security, privacy and cost are the primary gaps in most existing cloud-IoT architectures. In this research we specifically address security and privacy issues in cloud based IoT architecture.

## IV. INTEGRATION OF SDN AND HYBRID IOT

The public and private clouds have their own set of rules for collecting, transferring, managing and processing data.

We propose to integrate the SDN with the hybrid cloud in the sensor layer of IoT data collection. The integration would fill the security gap between hybrid cloud computing and IoT from data collection to transfer and analysis using SDN at the device layer. It also allows us to tackle security challenges using integration of SDN and hybrid cloud architecture in an efficient way.

 ISBN: 978-1-61208-711-5 

Having an IoT hybrid cloud architecture mixed with SDN technology will address the security and privacy issues of IoT data (from collection to the analysis phase). This architecture will therefore be a significant improvement in IoT technology and will encourage enterprise clients moving towards IoT technology. The presented approach will minimise the chance of data leakage during data collection, transfer and analysis.

In this research, devices will be categorised as public and private devices. This categorisation can be varied for different cases; however, devices are considered private when the data generated by them is sensitive. For instance, personal communication devices, health-related devices, etc. Other devices are public devices such as entertainment devices, doors, windows, kitchen appliance. In our proposed SDN design, data collected from devices have to pass some security layers before they sit in the right platform. We address most of the hybrid cloud IoT issues using SDN at the device layer. We isolate and encrypt private data before its arrival in the private cloud.

## V. CASE STUDY

In this section, we will illustrate the integration and the proposed solution for security. The challenge is that different network rules apply to sensor devices from the IoT side and to those in the data collection in the hybrid cloud architecture.

In our proposed solution, Figure 2 represents the existing collection network for IoT data. Figure 3 represents an edge node configuration that integrates an SDN controller with an OpenFlow-enabled switch. For the test bed, as an initial experiment, we are collecting environmental data in the form of indoor temperature, humidity, CO2, and outdoor data from a weather station including wind speed, outdoor temperature, pollen and dust count. The edge node is also represented in Figure 2 as a point of demarcation for data arriving at the edge node.

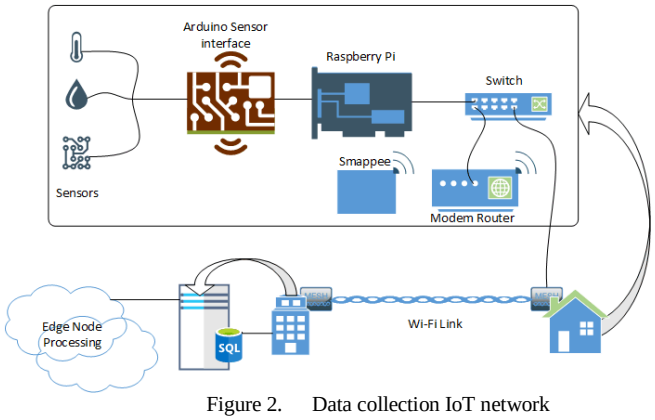



Figure 2. Data collection IoT network

Separating the collection of data represents a major area for this work. Collection examples from simple http header extraction locating embedded sensor serial numbers, to flow rules representing metadata and analog information from the devices to verify whether a transmission came from the expected transmitter in the expected location. In this way,

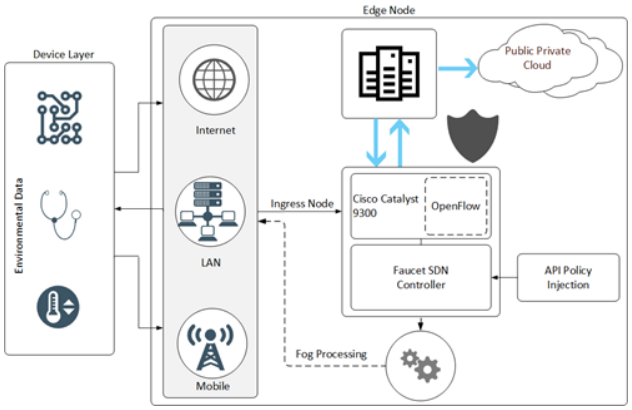



Figure 3. IoTaaS edge node design

proof of location and authentication from devices will provide a unique key. This combination of sensor data will provide the basis of flow tables to modify flows for the SDN controller.

Security between the buildings is handled by encrypting the point-to-point wireless link. All data arrives at the edge node having been collected from low power wireless (SigFox), 802.11, 4G or Ethernet.

Currently, site sensors represented in Figure 2, are hardwired through an Arduino, data is collected by a Raspberry Pi and sent point-to-point wirelessly between buildings. File transmission on the link is handled by Rsync. Encrypting at this point in the transmission network rather than at the sensor level takes a processing load off the sensor physical layer and ensures no additional burdens are placed on low powered sensors.

As data arrives at the Catalyst 9300 switch, OpenFlow match rules will segregate the data based on sensor location and the metadata generated by sensor hardware characteristics. Data will be sent to matching egress ports, depending on match and action rules. Some data will take the return path for correlation or further processing. In this phase, further processing will only be necessary for real time processes.

In the proposed solution, custom flow matches with the SDN controller, use the Openflow Extensible Match (OXM) and leverage the experimenter field.

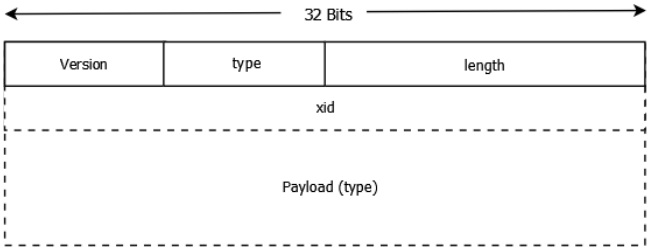



Figure 4. OpenFlow Header

Use of the experimenter field in OpenFlow requires the access to a vendor ID and is represented as class 0xfff which extends the header to 64 bits by using the first 32 bits of the

body as an experimenter field. Figure 4 shows the OpenFlow header. The experimenter field addition allows for matching of unknown and custom tables. For this work, matching criteria is based unique sensor metadata and characteristics.

As part of policy development, refinement and extending the range of rules based on flow matches within the OpenFlow controller, utilising external packet matching filtering will play a major role in the future of this project. The Berkeley Packet Filter (BPF) is one such set of filters able to optimise hardware ASICs [19].

Development of policy to optimise match rules already supported in OpenFlow V1.5 is focused on segregation of flows at the collection point for field networks Internet, LAN and Mobile. Match tables with corresponding flow instruction fields will separate data on interface, VLAN or both, as actions in response to flow matches.

Adding an experimenter OXM extension to the match fields of the SDN controller leverages the pipeline sequence processing employed by the OpenFlow protocol. Unique flows are identified by the combination of priority and match fields.

Flow entry instructions and action lists make it possible to pass packets to other flow tables or perform an action without further processing. This process could include re-writing packet headers in preparation for alternate egress ports based on flow type.

## VI. CONCLUSION AND FUTURE WORK

In this paper, the idea of integrating an SDN solution for managing Hybrid IoT data is presented. The aim is to use SDN to supervise efficient and secure Hybrid Cloud Computing to manage data collected by devices over the Internet. This design is leveraging the advantage of using Hybrid Cloud computing capability, storage and networking capability.

As a result, IoT will benefit from the performance, security and scalability of Hybrid Cloud Computing [1] while data collection and storage are managed by a secure network.

The emerging IoT paradigm challenges current network methods and practices. Network perimeters are potentially defined by the distribution of field devices deployed in homes, factories, agriculture and on-person. Beyond the issues of dealing with the flood of data generated from smart devices, addressing privacy and security is a priority for research. Private data traversing multiple network and storage domains pose perplexing issues for the integration of cloud computing and IoT.

Exponential growth of the IoT phenomenon has created a gap in the management of incoming-data processing. Furthermore, the inability to manage big data efficiently exacerbates the development of security processes for isolation of private sensitive data.

## ACKNOWLEDGMENT

We would like to thank Ara Institute of Canterbury for their support with providing the testing resources in the Cisco Networking Academy Lab during preparation for this paper.

## REFERENCES

[1] G. Fortino, A. Guerrieri, W. Russo and C. Savaglio, "Integration of agent-based and Cloud Computing for the smart objects-oriented IoT," Proceedings of the IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hsinchu, pp. 1-6, 2014.

[2] A. Botta, W. de Donato, V. Persico and A. Pescapé, "On the Integration of Cloud Computing and Internet of Things," International Conference on Future Internet of Things and Cloud, Barcelona, pp. 23-30, 2014.

[3] Q. Erwa, L. Yoanna, Z. Chenghong and H. Lihua, "Cloud Computing and the Internet of Things: Technology Innovation in Automobile Service", Yamamoto, Sakae, Berlin, Heidelberg, pp. 173-180, 2013.

[4] B.B. P. Rao, S. Payal, N. Sharma, A. Mittal and S.V. Sharma. "Cloud computing for Internet of Things & sensing based applications". 2012 Proceedings of the International Conference on Sensing Technology, ICST. pp. 374-380, 10.1109/ICSensT, 2012.

[5] S. Muhammad and S.Tariq. "Cyber Security and Internet of Things", 2017 [Online: last accessed February 2019].

[6] https://www.rackspace.com/library/what-is-a-bare-metal-server. 2018, [Online: last accessed March 2019].

[7] A. Sharma, et al., "A Secure Hybrid Cloud Enabled architecture for Internet of Things," IEEE 2nd World Forum on Internet of Things (WF-IoT), Milan, pp. 274-279, 2015.

[8] A. Sharma, E. S. Pilli and A. P. Mazumdar, "Obviating capricious behavior in internet of things," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, pp. 480-486, 2017.

[9] https://www.oracle.com/assets/2018-cloud-predictions, 2017, [Online: last accessed January 2019].

[10] J. Mineraud, O. Mazhelis, X. Su and S. Tarkoma, "A Gap Analysis of Internet of Things Platforms", Computer Communications, pp. 5-16, 2010.

[11] J. Mineraud, M. Oleksiy, S. Xiang and T. Sasu, "Contemporary Internet of Things Platforms", pp. 1-6, 2015.

[12] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, 2016.

[13] E. Hesham, et al., "Edge of Things: The Big Picture on the Integration of Edge", IoT and the Cloud in a Distributed Computing Environment. IEEE Access. pp. 1-1, 2017.

[14] D. Puthal, S. Nepal, R. Ranjan and J. Chen, "Threats to Networking Cloud and Edge Datacenters in the Internet of Things," in IEEE Cloud Computing, vol. 3, no. 3, pp. 64-71, 2016.

[15] H. Yoon, S. Kim, Taekho Nam and J. Kim, "Dynamic flow steering for IoT monitoring data in SDN-coordinated IoT-Cloud services," International Conference on Information Networking (ICOIN), Da Nang, pp. 625-627, 2017.

[16] R. Vilalta, et al., "End-to-End SDN/NFV Orchestration of Video Analytics Using Edge and Cloud", OFC, 2017.

[17] R. Muñoz, et al., "IoT-aware Multi-layer Transport SDN and Cloud Architecture for Traffic Congestion Avoidance Through Dynamic Distribution of IoT Analytics", Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), pp. 1-3, 2016.

[18] W. Dai, et al., "TNGuard: Securing IoT Oriented Tenant Networks based on SDN", IEEE Internet of Things Journal, vol. 5, no. 3, pp. 1-13, 2018.

[19] S. Jouet, R. Cziva and D. P. Pezaros, "Arbitrary packet matching in OpenFlow," 2015 IEEE 16th International Conference on High Performance Switching and Routing (HPSR), Budapest, pp. 1-6, 2015.

 ISBN: 978-1-61208-711-5 

# Optimal Energy Price-Aware Resource Allocation Scheme for Scheduled Lightpath Demands

Karan Neginhal, Saja Al Mamoori, Arunita Jaekel
School of Computer Science
University of Windsor
Windsor, Canada
Email: {neginha, sajak, arunita}@uwindsor.ca

***Abstract*—Electricity costs comprise a significant portion of operating expenses forData Center Networks (DCNs). As a result, energy-aware Routing and Wavelength Assignment schemes (RWA), which try to minimize the overall energy consumption for data transmission, have received considerable attention in the past decade. Recently, the idea of minimizing the dollar cost of energy consumption using Real-Time Pricing (RTP) has been proposed for Wavelength-Division Multiplexing (WDM) optical networks. The RTP-based RWA has been shown to result in reduced electricity costs. In this paper, we present a new formulation for optimal RWA for scheduled lightpath demands using the RTP model. Our results indicate that the proposed approach clearly outperforms the Flat-Rate Price (FRP) model, as well as traditional shortest path routing schemes.**

***Keywords–Data Center Networks (DCNs); Energy-aware resource allocation; Routing and Wavelength Assignment (RWA); Real-Time Price (RTP).***

## I. INTRODUCTION

Data Center Networks (DCNs) are one of the fastest growing consumers of electricity due to the rapid increase of digital content, big data, e-commerce and Internet traffic [1]-[5]. The electricity costs comprise a significant portion of operating expenses for such networks [6]-[10]. To mitigate this problem, the development of energy efficient schemes is crucial at all levels of network infrastructure, including data transmission. Many Routing and Wavelength Assignment (RWA) schemes that minimize energy consumption at network nodes and/or fiber links have been proposed in the literature for data transfer over an optical network [11]-[13]. However, energy prices using the RTP model can vary widely, depending on geographic location. Therefore, minimizing energy consumption may not necessarily result in the least dollar cost. Wavelength-Division Multiplexing (WDM) in optical networks is the technology that allows multiplexing a number of optical carrier signals onto a single optical fiber by using different wavelengths. In this study, we propose an energy-aware RWA optimal algorithm, which aims to minimize the dollar cost in WDM optical networks using RTP the model.

In recent years, various research works have been published in the field of energy efficient WDM networks. A number of different approaches have been proposed including reducing Electrical-Optical-Electrical (E-O-E) conversions [14], switching off or slowing down unused network elements [15][16] putting selected network components in sleep mode [17], and using intelligent traffic grooming techniques [18][19]. Energy aware unicast routing in WDM networks has received considerable research attention in the last ten years [20].

In many applications, the physical location of the server or other network resources remains hidden from the user as it is not important. In this scenario, it is possible to select the best destination from the set of possible destinations to execute a job. This is known as *anycast routing* [21]. This allows the routing algorithms the flexibility of choosing a suitable processing (destination) node for a given task, such that network resources can be utilized as efficiently as possible. Both heuristics and optimal formulations energy-aware approaches using anycast routing have been considered in [22]-[24]. The goal is to reduce both the static and the dynamic (load dependent) portions of power consumption as much as possible, although static power consumption typically dominates for most network components [24].

In WDM optical networks, there are mainly three different demand allocation models:

1) **Static traffic model**: Where the set of demands is fixed and known in advance.
2) **Dynamic traffic model**: Where the start time and the end time of the demands are known in advance. The set of demands is generated based on certain distributions.
3) **Scheduled traffic model**: Where the set of demands is predictable and periodic in nature.

In Scheduled Traffic Demands (STDs), the setup time and the tear down time for the demand is known in advance. The Scheduled Traffic Model (STM) is further divided into two different models, known as *fixed window traffic* model and *sliding scheduled traffic* model. A number of recent papers have shown how *anycast* routing can be used for minimizing the overall energy consumption in optical networks [23]-[25]. However, these papers mostly deal with the static [26][27] or dynamic [12][13] traffic models. In our previous work, we have considered energy-aware routing and traffic grooming of sub-wavelength demands, under STM [28]. Although energy aware routing for WDM networks has received significant attention in recent years, the idea of utilizing the anycast concept for energy minimization [16][23][25] has been less well studied.

Replication in DCNs makes it possible to have multiple copies of data on different Data Centers (DCs) [29]. Adding more replicas improves reliability, lowers latency across the network, and allows more flexibility in choosing energy efficient routes; but it also increases the costs for network equipment and storage [30]. Very recently, researchers have proposed reducing the operational expenditures by choosing the route with the least cost for the energy consumed based on Real-Time Pricing (RTP) [31]. They considered price changes throughout the day and for different US time zones. The Least Dollar Path (LDP) approach in [31] considers the real-time energy costs and replicated data storage to avoid costly peak charges and reduce the overall energy cost. The flat rate pricing

 ISBN: 978-1-61208-711-5 

model leads to more electricity costs as compared to the real-time pricing model [31]. Efficient routing schemes and proper arrangement of the replicas can lower energy consumption in the DCNs [31].

In this paper, we present a new Integer Linear Program (ILP) formulation for RTP-based optimal RWA of scheduled lightpath demands. Under STM, the setup and teardown times of the demands are known in advance, so that the RWA algorithm can optimize resource allocation in both space and time [32]. The proposed ILP not only selects the appropriate data center node to serve each request, but also performs RWA that leads to the least dollar cost. A heuristic algorithm for solving this problem has been presented in [33]. To the best of our knowledge, RTP-based energy-aware RWA for advance reservation under the fixed-window STM has not been considered before. Our approach differs from the previous RTP-based RWA as follows:

- We consider energy consumption not only at network nodes, but along fiber links as well.
- We process the set of demands as a whole, rather than adopt a greedy approach where each demand is processed one at a time.
- We consider both static and dynamic components of power consumption of nodes and links.

The remainder of the paper is organized as follows. In Section II, we outline our network energy model and propose an optimal formulation for energy-aware routing. In Section III, we present and analyze our simulation results and discuss our conclusions with some directions for future work in Section IV.

## II. Energy Efficient Anycast Routing For Fixed Window Scheduled Traffic Model

In this section, we introduce the proposed optimal algorithm formulated as an ILP using the anycast principle for fixed window scheduled lightpath demands allocation. The objective here is to minimize the overall electricity costs of a DCN by reducing the actual energy consumption.

### A. Network Energy Model

We consider a transparent IP-over-WDM network, which consists of optical cross connect switches (OXCs) connected to an IP router [34]. We consider power consumption both at network nodes and fiber links [24]. The total power consumption of the IP router and optical switch can be calculated using the following equations.

$$P_{IP} = P^s_{IP} + \pi_{IP} * t_{IP} \tag{1}$$

$$P_{SW} = P^s_{OXC} + \pi_{OXC} * t_{\lambda} \tag{2}$$

In (1) and (2), $P^s_{IP}$ and $P^s_{OXC}$ denote the static power consumption for the IP router and the optical switch, respectively. Similarly, $\pi_{IP}$ and $\pi_{OXC}$ denote the dynamic, i.e. traffic dependent, power consumption for the IP router and the optical switch, respectively. The terms $t_{IP}$ and $t_{\lambda}$ indicate the amount of traffic flowing through the IP router and the switch, respectively.

TABLE I. POWER CONSUMPTION OF NETWORK DEVICES [35][36].

| Device | Symbol | Power Consumption |
|---|---|---|
| IP router (static) | $P^s_{IP}$ | 150 W |
| OXC (static) | $P^s_{OXC}$ | 100 W |
| IP router (dynamic) | $\pi_{IP}$ | 17.6 W |
| OXC (dynamic) | $\pi_{OXC}$ | 1.5 W |
| Transponder (dynamic) | $\pi_{XT}$ | 34.5 W |
| Pre-amplifier | $P_{pre}$ | 10 W |
| Post-amplifier | $P_{post}$ | 20 W |
| Inline-amplifier | $P_{inline}$ | 15 W |

The power consumption of a link is obtained using (3), where $P_{pre}$, $P_{post}$ and $P_{inline}$ are the power consumed by pre, post, and inline amplifiers, respectively. The actual values of these parameters, used in our simulations, are taken from [35] and [36] and shown in TABLE I.

$$P_e = P_{pre} + P_{post} + P_{inline} \tag{3}$$

### B. Solution Approach

We consider a set of fixed window lightpath demands and propose a *minimum cost path* using RTP (MCP-RTP) model to select the route and destination for each demand in such a way that the overall electricity costs are minimized. The notation used in our ILP is given below.

### C. Notation used in this paper

$\boldsymbol{G(N, E)}$: Physical topology, where $N$ is set of nodes and $E$ is the set of bidirectional edges (i.e., links) in the network.
$\boldsymbol{N}$: Set of data center nodes.
$\boldsymbol{(i, j)}$: An edge in the network from node $i$ to node $j$.
$\boldsymbol{Q}$: Set of lightpath demands to be routed over the physical topology. Each demand is a tuple $(s_q, st_q, \tau_q)$, where $s_q$ is the source node for demand $q$, $st_q$ is the starting time for demand $q$ and $\tau_q$ denotes the holding time for demand $q$.
$\boldsymbol{m}$: $= 0, 1, 2, ...m_{max}$, where $m$ is the number of intervals ($0 \leq m \leq 23$).
$\boldsymbol{a_{q,m}}$: = 1 if demand $q$ is active during interval $m$.
$\boldsymbol{l_e}$: length of edge $e$.

*Binary Variables*
$\boldsymbol{IP_{i,m}}$: = 1, if the IP router at node $i$ is active at interval $m$.
$\boldsymbol{OXC_{i,m}}$: = 1, if OXC at node $i$ is active at interval $m$.
$\boldsymbol{L_{e,m}}$: = 1, if link $e$ is in use at interval $m$.
$\boldsymbol{x_{q,e}}$: = 1, if lightpath $q$ uses link $e$.
$\boldsymbol{y_{q,i}}$: = 1, if lightpath $q$ uses node $i$.
$\boldsymbol{dc_{q,i}}$: = 1, if DC node $i$ is selected as a destination for lightpath $q$.

*Bounded Variables*
$\boldsymbol{\beta^q_{i,m}}$: = 1, if lightpath $q$ uses IP router at node $i$ during interval $m$.
$\boldsymbol{\gamma^q_{i,m}}$: = 1, if lightpath $q$ uses OXC at node $i$ during interval $m$.

$\boldsymbol{\sigma^q_{e,m}}$: = 1, if lightpath $q$ uses link $e$ during interval $m$.

$$minimize \sum_m \Bigg[ \sum_i cost_{i,m} \Bigg[ P^s_{IP} \dot{+} \pi_{IP} \sum_q \beta^q_{i,m} + \left( P^s_{i,m} \dot{O}XC_{i,m} \pi_{OXC} \sum_q \gamma^q_{i,m} \right) + \left( \pi_{XT} \sum_q \beta^q_{i,m} \right) \Bigg] + cost_{j,m} \sum_{e:(i,j)} P_e \dot{L}_{e,m} \Bigg]$$

$$\sum_{e:(i,j)\in E} x_{q,e} - \sum_{e:(j,i)\in E} x_{q,e} = \begin{cases} dc_{q,i}, \text{if } i = source, \\ -dc_{q,i}, \text{if } i = destination, \\ 0, \text{otherwise.} \end{cases} \tag{4}$$

Constraint (4) must be satisfied $\forall i \in N, q \in Q$

$$y_{q,i} = \sum_{e:(i,j)\in E} x_{q,e} \quad \forall i \in N, q \in Q \tag{5}$$

$$\sum_q x_{q,e} \cdot a_{q,m} \leq |K| \quad \forall e \in E, 1 \leq m \leq m_{max} \tag{6}$$

$$\sum_{i \in S} dc_{q,i} = 1 \quad \forall q \in Q; \quad dc_{q,i} = 0 \quad \forall i \notin S, q \in Q \tag{7}$$

IP router usage:

$$dc_{q,i} + a_{q,m} - \beta^q_{i,m} \leq 1 \tag{8}$$

$$dc_{q,i} \geq \beta^q_{i,m} \tag{9}$$

$$a_{q,m} \geq \beta^q_{i,m} \tag{10}$$

$$IP_{i,m} \geq \beta^q_{i,m} \tag{11}$$

$$IP_{i,m} \leq \sum_q \beta^q_{i,m} \tag{12}$$

Constraints (8) - (12) must be satisfied $\forall i \in S, q \in Q, 1 \leq m \leq m_{max}$.

OXC switch usage:

$$(dc_{q,i} + y_{q,i}) + a_{q,m} - \gamma^q_{i,m} \leq 1 \tag{13}$$

$$(dc_{q,i} + y_{q,i}) \geq \gamma^q_{i,m} \tag{14}$$

$$a_{q,m} \geq \gamma^q_{i,m} \tag{15}$$

$$OXC_{i,m} \geq \gamma^q_{i,m} \tag{16}$$

$$OXC_{i,m} \leq \sum_q \gamma^q_{i,m} \tag{17}$$

Constraints (13) - (17) must be satisfied $\forall i \in N, q \in Q, 1 \leq m \leq m_{max}$.

Link usage:

$$x_{q,e} + a_{q,m} - \sigma^q_{e,m} \leq 1 \tag{18}$$

$$x_{q,e} \geq \sigma^q_{e,m} \tag{19}$$

$$a_{q,m} \geq \sigma^q_{e,m} \tag{20}$$

$$L_{e,m} \geq \sigma^q_{e,m} \tag{21}$$

$$L_{e,m} \leq \sum_q \sigma^q_{e,m} \tag{22}$$

Constraints (18) - (22) must be satisfied $\forall e \in E, q \in Q, 1 \leq m \leq m_{max}$.

### *D. Justification of the ILP*

The objective function tries to minimize the dollar cost by using the real time electricity prices. The summation is over all intervals $m$ and for each network component, i.e., IP router, optical switch and fiber link. The term $cost_{i,m}$ is the real time electricity price at node $i$ during interval $m$. We have 24 intervals and for each interval the electricity price is different. For calculating the cost of a link $e : i \rightarrow j$, we have multiplied the energy consumption of the link $e$ with the RTP electricity cost at the destination node $j$ of that link.

Constraint (4) is the standard flow conservation constraint, which finds a feasible path from source node $s_q$ to the selected data center (destination) node $dc_{q,i}$ for each demand $q$. Constraint (5) ensures that if lightpath $q$ traverses link $e : i \rightarrow j$ the value of $y_{q,i}$ is set to 1. Constraint (6) ensures that the total number of demands traversing link $e : i \rightarrow j$ does not exceed the number of available channels $|K|$. Constraint (7) ensures that exactly one data center is selected as the destination node for lightpath $q$.

Constraints (8) - (12) are the *IP router usage* constraints. They are used to determine if a particular IP router at node $i$ is active during interval $m$. Constraints (8) - (10) are used to set the value of $\beta^q_{i,m}$. Constraint (8) sets $\beta^q_{i,m}$ to 1 if lightpath $q$ is active during interval $m$ and DC node $i$ is selected as a destination for lightpath $q$. Constraints (9) and (10) ensure that $\beta^q_{i,m}$ is set to 0 if either $dc_{q,i}$ or $a_{q,m}$ is 0. Constraint (11) ensures that if the IP router is active at node $i$ during interval $m$ it is used by at least one lighthpath $q$. Constraint (12) ensures that if there is no lightpath $q$ using the IP router at node $i$ during interval $m$ then the IP router is not active during that interval $m$, i.e., $IP_{i,m}$ = 0.

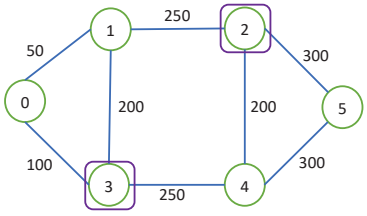
Constraints (13) - (17) are the *optical switch usage* constraints. They are used to determine if a particular optical

switch at node $i$ is active during interval $m$. Constraints (13) - (15) are used to set the value of $\gamma^q_{i,m}$. Constraint (13) sets $\gamma^q_{i,m}$ to 1 if lightpath $q$ is active during interval $m$ and uses the OXC at node $i$. Constraints (14) and (15) ensure that $\gamma^q_{i,m}$ is set to 0, if either $dc_{c,q} + y_{q,i}$ or $a_{q,m}$is 0 . Constraint (16) ensures that the OXC switch is active at node $i$ during interval $m$ if it is used by at least one lightpath $q$. Constraint (17) ensures that if there is no lightpath $q$ using OXC switch at node $i$ during interval $m$, then the OXC switch is not active during that interval $m$, i.e., $OXC_{i,m}$= 0.

Constraints (18) - (22) are the *link usage* constraints. They are used to determine if a particular link is active during interval $m$. Constraints (18) - (20) are used to set the value of $\sigma^q_{e,m}$. Constraint (18) sets $\sigma^q_{e,m}$ to 1 if lightpath $q$ uses link $e$ and is active during interval $m$. Constraints (19) and (20) ensure that $\sigma^q_{e,m}$ is set to 0 if either $x_{q,e}$ or $a_{q,m}$is 0 . Constraint (21) ensures that link $e$ is active during interval $m$ if it is used by at least one lightpath $q$. Constraint (22) ensures that if there is no lightpath $q$ using link $e$ during interval $m$, then the link is not active during that interval $m$, i.e., $L_{e,m}$ = 0.

### E. An Illustrative Example

To illustrate the effectiveness of the proposed approach, we consider a simple 6-node network with 8 bi-directional links. The physical topology used in this example is shown in Figure 1a. The label on each edge represents the length of the link in Km. Nodes 2 and 3 are identified as the data center nodes, which will serve as potential destinations for the connection (lightpath) demands.




| | $s_q$ | $st_q$ | $\tau_q$ |
|---|---|---|---|
| $LP_0$ | 1 | 5 | 8 |
| $LP_1$ | 4 | 4 | 15 |
| $LP_2$ | 5 | 0 | 7 |

(b)

Figure 1. (a) A sample physical topology and (b) A sample set of lightpath demands.

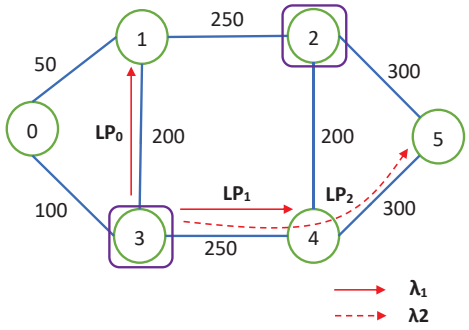



Figure 2. Routing of lightpath demands for the proposed objective.

A set of 3 lightpath demands is shown in Figure 1b, where $s_q$ indicates the source node, $st_q$ indicates the starting time interval for that demand and $\tau_q$ indicates the holding time for the demand, in terms of the number of time intervals. For example, according to the lightpath requests table, the lightpath LP0 originates from node 1, at interval 5 and is active for a total of 8 intervals. Based on our main objective, which tries to minimize the dollar costs by reducing the power consumption in the DCNs, the ILP selects the appropriate destination (i.e., data center node) and finds the "best" route with minimum dollar cost to the selected destination.

Figure 2 shows the routing scheme of lightpath demands on the given physical topology based on our objective, which minimizes the overall dollar cost. To explain how the lightpaths are routed based on our approach's minimum dollar cost objective, we consider the following examples:

- lightpath LP1 is using the route $3 \rightarrow 4$ where the selected data center is node 3. The approach could have chosen an alternative data center at node 2 if the objective was to minimize the distance, for instance.
- Similarly, lightpath LP2 is using the route $3 \rightarrow 4 \rightarrow 5$ with data center node 3 as the destination based on our objective. If the objective was to minimize the path distance or the number of hops, for example, then the ILP could have chosen the route $2 \rightarrow 5$ with data center node 2 as the destination instead.

## III. Simulation Results

For our simulations, we consider three well-known topologies: the 11-node COST-239, the 14-node NSFNET, and 24-node USANET [20]. The number of lightpath demands used in the simulations ranged from 40 to 120. The holding time of each demand ranged from 4 hours to 15 hours, with an average duration of 5 hours. The results reported in this section correspond to average values over 5 different runs. The simulation was carried out with IBM ILOG CPLEX 12.6.2.

Results are reported for four different approaches listed below, for different networks and traffic loads.

- The proposed ILP (*MCP-RTP*)
- The minimum hop path (*MHP*)
- The shortest distance path (*SDP*)
- The mininum cost path with flat rate pricing (*MCP-FRP*)

The dollar costs for routing 40 demands over different topologies and for three different approaches, our proposed approach, MCP-RTP, MHP, and SDP are shown in Figure 3. As seen in the figure, our main approach, which minimizes the electricity cost, has the lowest dollar cost for all cases, as expected. The improvement ranges from 36% - 62.9% compared to MHP approach which aims to minimize the path number of hops and 31.8% - 63.4% compared to SDP which minimizes the path distance.

A comparison of dollar costs for routing different demands over the 14-node topology, using the three approaches of Figure 3, is shown in Figure 4. A standard growth in the dollar cost values is observed with an increase in the demand size. As expected, our proposed approach MCP-RTP, performs better than the other approaches in reducing the dollar cost.

 ISBN: 978-1-61208-711-5 

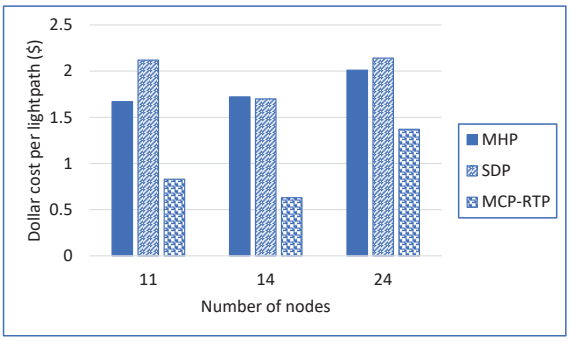



Figure 3. Comparison of electricity costs with different RWA approaches for different topologies and 40 demands

The improvement ranges from 32.9% - 63% over the MHP and 39.3% - 63.4% over SDP for all traffic loads.

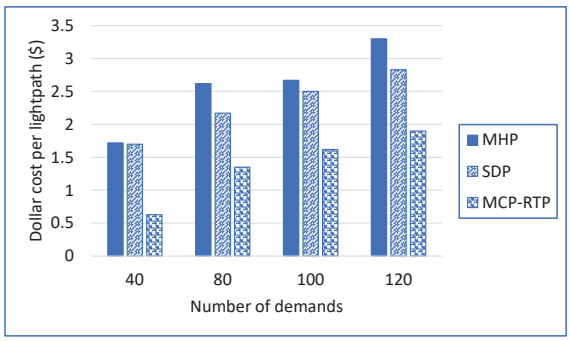



Figure 4. Comparison of electricity costs with different RWA approaches and different demands for NSFNET network

In Figure 5, a comparison of the overall dollar cost is shown for routing 40 demands over the different topologies with our approach MCF-RTP and the MCF-FRP approach, which tries to minimize the dollar cost using the flat-rate price model. Our approach outperforms the MCF-FRP by reducing the electricity cost by an average of 60%.

We illustate how the overall cost varies with the number of demands for the 14-node NSFNET topology, for MCP-RTP and MCP-FRP routing schemes in Figure 6. As expected, the proposed method (MCP-RTP) clearly outperforms the others, with an average reduction of 53% in cost. We note that this reduction in cost comes at the expense of slightly longer paths for routing demands, in some cases.

## IV. Conclusion

In this paper, we have proposed an ILP for the RTP energy-aware RWA for the fixed window scheduled traffic model. We have considered the anycast routing scheme to select the best option for the destination node and the real-time pricing model for selecting lightpaths' routes. The objective of this model is to reduce the overall electricity cost by reducing the actual power consumption and using nodes and links with lower costs. Our simulation results indicate that the proposed

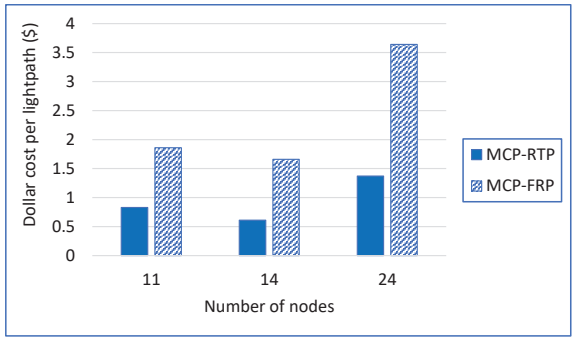



Figure 5. Comparison of electricity costs between MCF-RTP and MCF-FRP for different topologies and 40 demands

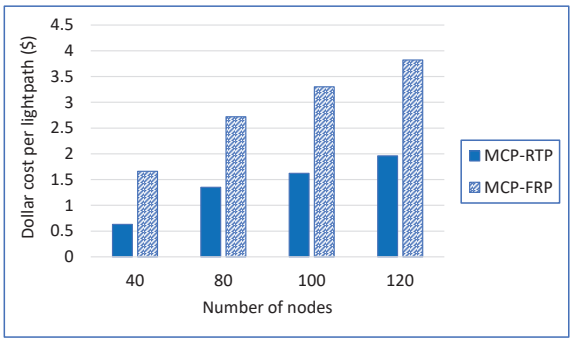



Figure 6. Comparison of electricity costs between MCF-RTP and MCF-FRP with different traffic loads for NSFNET

approach results in significant reductions in electricity costs, compared to both flat-rate pricing and traditional shortest-distance or minimum-hop routing.

In this work, we have primarily focused on energy costs. In the future, it will be interesting to incorporate other Quality of Service (QoS) metrics, such as bandwidth and delay into our model and evaluate the performance in terms of these metrics. It is also worthwhile to consider trade-offs of selecting the least cost path, which may have higher energy consumption, and compare the results with existing works that minimize energy consumption. Finally, this work can be extended to consider the sliding STM, so demand start times can be optimally adjusted, to further reduce energy costs.

## Acknowledgment

The work of A. Jaekel has been supported by a research grant from the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

[1] A. Deylamsalehi, Y. Cui, P. Afsharlar, and V. M. Vokkarane, "Minimizing electricity cost and emissions in optical data center networks," Journal of Optical Communications and Networking, vol. 9, no. 4, 2017, pp. 257–274.

[2] H.-P. Jiang, D. Chuck, and W.-M. Chen, "Energy-aware data center networks," Journal of Network and Computer Applications, vol. 68, 2016, pp. 80–89.

 ISBN: 978-1-61208-711-5 

[3] Y. Lui, G. Shen, and W. Shao, "Energy-minimized design for ip over wdm networks under modular router line cards," in 2012 1st IEEE International Conference on Communications in China (ICCC), IEEE 2012, pp. 266–269.

[4] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in ACM SIGCOMM computer communication review, vol. 39, no. 4, ACM, 2009, pp. 123–134.

[5] S. Yang, P. Wieder, R. Yahyapour, and X. Fu, "Energy-aware provisioning in optical cloud networks," Computer Networks, vol. 118, 2017, pp. 78–95.

[6] A. Ebrahimzadeh, A. G. Rahbar, and B. Alizadeh, "Pli-aware cost management for green backbone all-optical wdm networks via dynamic topology optimization," IEEE/OSA Journal of Optical Communications and Networking, vol. 10, no. 9, 2018, pp. 785–795.

[7] D. Rami, Energy Efficient Anycast Routing for Sliding Scheduled Lightpath Demands in Optical Grids. University of Windsor, 2016.

[8] S. Al Mamoori, D. Rami, and A. Jaekel, "Energy-efficient anycast scheduling and resource allocation in optical grids," Journal of Ambient Intelligence and Humanized Computing, vol. 9, no. 1, 2018, pp. 73–83.

[9] X. Dong, T. El-Gorashi, and J. M. Elmirghani, "Green ip over wdm networks with data centers," Journal of Lightwave Technology, vol. 29, no. 12, 2011, pp. 1861–1880.

[10] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–9.

[11] K. Manousakis, A. Angeletou, and E. Varvarigos, "Energy efficient rwa strategies for wdm optical networks," IEEE/OSA Journal of Optical Communications and Networking, vol. 5, no. 4, 2013, pp. 338–348.

[12] S. Ricciardi, F. Palmieri, U. Fiore, D. Careglio, G. Santos-Boada, and J. Solé-Pareta, "An energy-aware dynamic rwa framework for next-generation wavelength-routed networks," Computer Networks, vol. 56, no. 10, 2012, pp. 2420–2442.

[13] A. Coiro, M. Listanti, and A. Valenti, "Dynamic power-aware routing and wavelength assignment for green wdm optical networks," in 2011 IEEE International Conference on Communications (ICC), IEEE, 2011, pp. 1–6.

[14] D. Tafani, B. Kantarci, H. T. Mouftah, C. McArdle, and L. P. Barry, "Distributed management of energy-efficient lightpaths for computational grids," in Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE, 2012, pp. 2924–2929.

[15] A. Coiro, M. Listanti, A. Valenti, and F. Matera, "Energy-aware traffic engineering: A routing-based distributed solution for connection-oriented ip networks," Computer Networks, vol. 57, no. 9, 2013, pp. 2004–2020.

[16] C. Develder, M. Tornatore, M. F. Habib, and B. Jaumard, "Dimensioning resilient optical grid/cloud networks," in Communication Infrastructures for Cloud Computing. IGI Global, 2014, pp. 73–106.

[17] A. Coiro, M. Listanti, A. Valenti, and F. Matera, "Reducing power consumption in wavelength routed networks by selective switch off of optical links," IEEE Journal of Selected Topics in Quantum Electronics, vol. 17, no. 2, 2011, pp. 428–436.

[18] M. M. Hasan, F. Farahmand, and J. P. Jue, "Energy-awareness in dynamic traffic grooming," in Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC). IEEE, 2010, pp. 1–3.

[19] S. Zhang, D. Shen, and C.-K. Chan, "Energy-efficient traffic grooming in wdm networks with scheduled time traffic," Journal of Lightwave Technology, vol. 29, no. 17, 2011, pp. 2577–2584.

[20] G. Shen and R. S. Tucker, "Energy-minimized design for ip over wdm networks," Journal of Optical Communications and Networking, vol. 1, no. 1, 2009, pp. 176–186.

[21] C. Develder, B. Dhoedt, B. Mukherjee, and P. Demeester, "On dimensioning optical grids and the impact of scheduling," Photonic Network Communications, vol. 17, no. 3, 2009, pp. 255–265.

[22] J. Buysse, C. Cavdar, M. De Leenheer, B. Dhoedt, and C. Develder, "Improving energy efficiency in optical cloud networks by exploiting anycast routing," in Communications and Photonics Conference and Exhibition, 2011. ACP. Asia. IEEE, 2011, pp. 1–6.

[23] Y. Chen and A. Jaekel, "Energy optimization in optical grids through anycasting," in Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013, pp. 3835–3839.

[24] Y. Chen, A. Jaekel, and K. Li, "Energy efficient anycast routing for scheduled lightpath demands in optical grids," in Communications (QBSC), 2014 27th Biennial Symposium on. IEEE, 2014, pp. 10–13.

[25] B. G. Bathula and J. M. Elmirghani, "Energy efficient optical burst switched (obs) networks," in GLOBECOM Workshops, 2009 IEEE. IEEE, 2009, pp. 1–6.

[26] K. Zhu and B. Mukherjee, "Traffic grooming in an optical wdm mesh network," in Communications, 2001. ICC 2001. IEEE International Conference on, vol. 3. IEEE, 2001, pp. 721–725.

[27] K. Lee and M. A. Shayman, "Optical network design with optical constraints in ip/wdm networks," IEICE transactions on communications, vol. 88, no. 5, 2005, pp. 1898–1905.

[28] Y. Chen and A. Jaekel, "Energy aware resource allocation based on demand bandwidth and duration," Procedia Computer Science, vol. 10, 2012, pp. 998–1003.

[29] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," ACM SIGARCH Computer Architecture News, vol. 38, no. 3, 2010, pp. 338–347.

[30] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future generation computer systems, vol. 28, no. 5, 2012, pp. 755–768.

[31] A. Deylamsalehi, P. Afsharlar, and V. M. Vokkarane, "Real-time energy price-aware anycast rwa in optical data center networks," in Computing, Networking and Communications (ICNC), 2016 International Conference on. IEEE, 2016, pp. 1–6.

[32] B. Wang, T. Li, X. Luo, Y. Fan, and C. Xin, "On service provisioning under a scheduled traffic model in reconfigurable wdm optical networks," in Broadband Networks, 2005. BroadNets 2005. 2nd International Conference on. IEEE, 2005, pp. 13–22.

[33] M. KHODA, Heuristic for Lowering Electricity Costs for Routing in Optical Data Center Networks. University of Windsor, 2018.

[34] K. Christodoulopoulos, K. Manousakis, and E. Varvarigos, "Offline routing and wavelength assignment in transparent wdm networks," IEEE/ACM Transactions on Networking (TON), vol. 18, no. 5, 2010, pp. 1557–1570.

[35] F. Musumeci, M. Tornatore, and A. Pattavina, "A power consumption analysis for ip-over-wdm core network architectures," Journal of Optical Communications and Networking, vol. 4, no. 2, 2012, pp. 108–117.

[36] A. Coiro, M. Listanti, A. Valenti, and F. Matera, "Power-aware routing and wavelength assignment in multi-fiber optical networks," Journal of Optical Communications and Networking, vol. 3, no. 11, 2011, pp. 816–829.


 ISBN: 978-1-61208-711-5 

# Least Loaded Sharing in Fog Computing Cluster

Sammy Chan
Department of Electronic Engineering
City University of Hong Kong
Hong Kong SAR
P.R. China
Email: eeschan@cityu.edu.hk

***Abstract*—Recently, there has been a growing ubiquity of connected devices and sensors in wireless sensor networks, health care systems, smart grids and smart cities, forming the Internet of Things (IoT). IoT devices generally have limited computation resources, and thus rely on the computational and storage resources of the cloud. However, IoT applications generally have a real-time requirement that cannot be fulfilled by mainstream cloud services. Therefore, a new paradigm called *fog computing* has emerged to offload the computation and storage needs of end user devices to the servers in the network edge. In this paper, we propose a least loaded sharing method to fully exploit the collaboration between fog servers and to achieve load balance among them. In our method, an overloaded server is able to react to temporary peaks of requests by forwarding the incoming requests to the least loaded neighbour server. The proposed method helps to reduce the blocking probability of requests and the delay experienced by accepted requests. We also develop a computationally efficient analytical model to evaluate the performance of our proposed method.**

***Keywords–fog computing; load balancing; collaboration servers; buffer sharing.***

## I. INTRODUCTION

Over the past decade, cloud computing has become a very popular computing paradigm [1]. By centralizing computing, storage, and network management functions in data centers, cloud computing has a high degree of polymerization of service computing. It enables end users to universally access on-demand computing services and frees them from the specification of many details. Entirely dependent on the Internet, cloud computing ensures the maximum utilization of computational resources by providing flexibility in the availability of data, software and infrastructure [2].

Recently, cloud computing is increasingly used to support Internet of Things (IoT) applications, due to the growing ubiquity of connected devices and sensors in wireless sensor networks, health care systems, smart grids and smart cities. Although cloud computing is renowned for its cost-effective and convenient service, it is encountering several challenges introduced by the emerging IoT. First, many IoT applications have a real-time requirement that cannot be fulfilled by mainstream cloud services [3][4]. Second, the vast and rapidly growing number of connected IoT devices inflate the amount of data generated at an exponential rate [5]. If all this data is sent to the cloud, prohibitively high network bandwidth would be required in the cloud system. Third, IoT devices generally have limited computation resources. Thus, they would not be able to fulfill the needs imposed by the IoT applications. Naturally, they could make use of the cloud by offloading computation tasks to it. However, it will be unrealistic and prohibitively expensive to support the interaction between the cloud and all those resource-constrained devices, as it involves complex protocols and resource-intensive processing.

To fill the technology gaps in supporting IoT, a new paradigm, *fog computing*, has been proposed. Fog computing emphasizes the network edge and distributes onerous tasks closer to end user devices [6][7]. As illustrated in Figure 1, fog computing extends cloud computing by bringing heterogeneous resources to the edge of the network, so that a substantial amount of data storage, computing and control functions, communication and networking is carried out near the end user. Besides, fog computing will not be faced with serious security issue as data travel from fog to end users within a short distance. Ultimately, the goals of fog computing are to reduce the data volume and traffic to cloud servers, offer low latency, and improve Quality of Service (QoS).

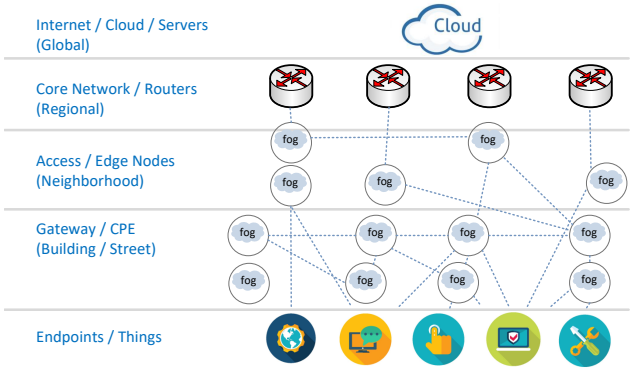



Figure 1. Architecture of fog computing.

Fog computing also shares many similar mechanisms and attributes with cloud computing [8]. For example, the core idea behind the two computing paradigms is to transfer load from end users to servers [9]. It means the problem of load imbalance is inevitable in both fog and cloud computing because requests of users arrive at servers randomly and frequently [10]. For the case of fog computing, servers with computing and storage capacities can be attached to the base stations of mobile telecommunication networks so that they are close to the end users. In the simplest case, an individual server only needs to execute tasks from its local users. In other words, each server operates independently. However, with the trend of deploying small cells, each server generally has some nearby neighbours. When a server experiences temporary overload, it could exploit the resources of its neighbours by forwarding

 ISBN: 978-1-61208-711-5 

newly arrived tasks to them. This is particularly attractive for fog computing, since it is designed to offer real-time service and thus needs to satisfy stringent QoS requirements, such as blocking of requests and service waiting time [11]. Under this situation where a set of servers collaborate with each other to serve users' requests, how to maintain load balance among the servers is an important issue.

In this paper, we propose a least loaded sharing method to fully exploit the collaboration between servers and achieve load balance among them. In our method, an overloaded server is able to react to temporary peaks of requests by forwarding the incoming request to the least loaded neighbour server. The proposed method helps to minimize the blocking probability of requests and reduce the delay experienced by accepted requests. We also develop a computationally efficient analytical model to evaluate the performance of our proposed method. The accuracy of the model is validated by computer simulations. Numerical results demonstrate that, by having servers share buffer space with each other, temporary load peaks can be efficiently relieved.

The remainder of the paper is organized as follows: Section II reviews some recent research work in optimization of computational resources in fog computing environments. Section III introduces our proposed least loaded scheme. Section IV presents the performance model of the scheme. Section V validates our model by simulation results and provides some numerical results to demonstrate the effectiveness of our proposed scheme. Finally, Section VI concludes the paper.

## II. Related Work

Fog computing allows mobile terminals to have access to additional computational and storage resources, which are more abundant than those available in typical user equipment, by offloading demanding tasks to nearby fog servers. For the case when each individual fog server only serves its local users, research efforts have focused on the joint distribution of computational and radio resources for mobile terminals and the fog server. Early work covered the management aspects, the experimental evaluation of energy saving due to offloading, and the design of offloading criteria which consider the cost of radio resources of the access networks [12][13]. Since the optimal energy cost for the data transfer depends on the channel conditions, the Gilbert-Elliott channel model is used in [14] to study the radio-cloud interaction. The work provides some insights about how the quality of the wireless link affects the transmission rate and the offloading decision. In [15], a computation offloading distribution between a single user and the server is derived, taking into account the delay constraint of tasks and assuming that multiple antennas are available. Their results provide the optimal transmission strategy and the optimal distribution of the computational load between the user and the server. When the number of requests from users becomes very large, the computational resources of only one server sometimes may not be enough. In this situation, a number of servers can cooperate together through the formation of a cluster. This means that extra computational capacities can be provided to users. In [16], Barbarossa *et al.* consider a multi-user, multi-server and multi-cloud scenario in which the servers and cloud are organized in a different hierarchy. They study a joint optimization of computational and radio resources with the objective to minimize the power consumption of each user, subject to the latency constraints imposed by each user.

When fog servers form a cluster, how to improve the QoS delivered to users is also an important issue. In [10], a load balancing scheme between two fog servers is proposed to minimize the blocking probability at each server and the waiting time of the tasks. In this scheme, each server is assumed to have a buffer to store service requests from users for subsequent executions. When the buffer of a server is full, the newly arrived requests are forwarded to the neighbour server, which accepts the request only if its current queue length is below a given threshold. The system is modelled as a two-dimensional Markov chain to evaluate the performance of the proposed scheme. Numerical results demonstrate that both blocking probability and waiting time are reduced. The authors also propose a possible implementation of the load balancing scheme.

## III. Least Loaded Sharing

We consider that, in an area of interest, there is a cluster of servers. Each server receives task requests from its local users, and executes the tasks on a first-come-first-serve basis. When a request arrives at a server and finds the server busy, it queues for its turn of service. Since the requests are expected to have strict delay requirements, there is a limit on the number of requests that can queue in a server. When the queue length of the server has reached the limit, for the case that each server operates independently, the request is blocked immediately to avoid excessive waiting time. On the other hand, when our least loaded sharing method is operated, the request is re-directed to the server with the shortest queue length. If there are multiple such servers, the request is re-directed to one of them randomly. As a result, a request is blocked only if the cluster of servers has reached the queue length limit.

## IV. Performance Models

Let us assume that there are $N$ servers, and task requests arrive at each server according to a Poisson process with rate $\lambda$ tasks/second, as shown in Figure 2. The time needed to execute a task is exponentially distributed with mean $1/\mu$ seconds. The offered traffic to each server is given by $\rho = \lambda/\mu$. The limit on the queue length (including the one being served) is set to $K$. Here, two metrics are used to evaluate the performance of our proposed method. The first one is the average waiting time of tasks, $\overline{W}$, which is the average of the time from the arrival of a task at a server until the time that the server starts processing the task. The second one is the task blocking probability, $p_B$, which is the probability that a task is blocked by the fog computing system.

First, we present the performance model for the *no-sharing* case, i.e., the case in which each server operates independently. This case will be used for comparison in the next section. For each independent server, it can simply be modelled as a $M/M/1/K$ queueing system. Let $\pi_i$, $0 \leq i \leq K$, be the probability that there are $i$ tasks in a server. For $\rho \neq 1$, it is well known that [18]

$$\pi_i = \rho^i \frac{1-\rho}{1-\rho^{K+1}}. \quad (1)$$

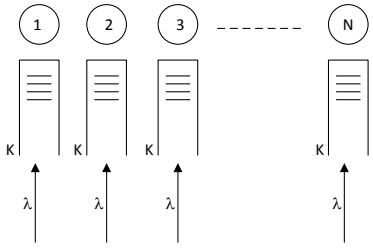



Figure 2. The queueing model of a cluster of fog servers.

The blocking probability is given by

$$p_B = \pi_K = \frac{\rho^K(1-\rho)}{1-\rho^{K+1}}, \tag{2}$$

and the mean waiting time is given by

$$\begin{aligned} \overline{W} &= \frac{\sum_{i=0}^{K} i\pi_i}{\lambda(1-\pi_K)} - 1/\mu \\ &= \left(\frac{1-(K+1-K\rho)\rho^K}{(1-\rho)^2} - 1\right)\frac{1}{\mu}. \end{aligned} \tag{3}$$

Next, we present the performance model for the least loaded sharing method. A server is said to be in state $i, 0 \leq i \leq K$, when there are $i$ tasks in its buffer (including the one being served). Let $p_i^j$ be the probability that server $j$ is in state $i$. However, since the system under consideration is uniform, each server receives the same offered load and has the same state probabilities. Thus, $p_i^j$ can be simplified as $p_i$. Consider a tagged server which is in state $i$ and has the shortest queue length. The probability that there are $l-1$ other servers at state $i$ is denoted as $F(l|i)$. Consider that a particular server is full. Its overflow rate to the tagged server at state $i$ is given by

$$\begin{aligned} y_i &= \lambda p_K \sum_{l=1}^{N-1} \tfrac{1}{l} F\left(l/i\right) \\ &= \lambda p_K \sum_{l=1}^{N-1} \tfrac{1}{l}\binom{N-2}{l-1} {p_i}^{l-1} \left(\sum_{t=i+1}^{K} p_t\right)^{N-1-l} \end{aligned} \tag{4}$$

Let $a_i$ be the total overflow rate of tasks to the tagged server when it is in state $i$,

$$a_i = (N-1)y_i \tag{5}$$

The tagged server can be modeled as a Markov chain, as shown in Fig 3. By using the local balance equation, we have

$$p_i = \frac{\lambda + a_{i-1}}{\mu} p_{i-1}, \quad i = 1, 2, \ldots, K \tag{6}$$

Therefore,

$$p_i = \frac{\prod_{j=0}^{i-1}(\lambda + a_j)}{\mu^i} p_0, \quad i = 1, 2, \ldots, K \tag{7}$$

Using the normalization condition $\sum_{i=0}^{K} p_i = 1$, $p_0$ is given by

$$p_0 = \left[\sum_{i=1}^{K} \frac{\prod_{j=0}^{i-1}(\lambda + a_j)}{\mu^i} + 1\right]^{-1}. \tag{8}$$

Equations (5) and (7) form a set of fixed-point equations. They can be solved by repeated substitution to obtain $p_i$.

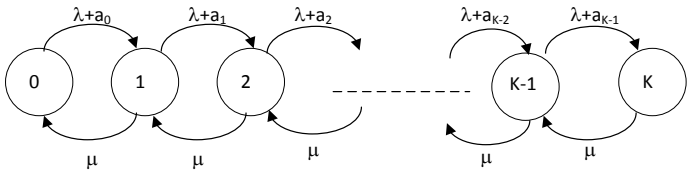



Figure 3. The Markov chain of a server.

By Little's theorem, $\overline{W}$ is given by

$$\overline{W} = \frac{\sum_{i=0}^{K} i p_i}{\lambda(1-p_K)} - 1/\mu \tag{9}$$

For $p_B$, at a first glance, one may think that it is given by $(p_K)^N$. However, the expression $(p_K)^N$ assumes that each server operates independently, which contradicts the fact that there is dependency among the servers. In order to obtain an exact value of $p_B$, we need to model the system as a $N$ dimensional Markov chain with totally $(K+1)^N$ states, and then solve the state probabilities. Unfortunately, efficient methods for solving the state probabilities are available only for very small $N$. For example, in [10], the matrix-geometric approach of [17] is used for the case of $N = 2$. For large $N$ or $K$, such an approach is analytically intractable. Here, we propose an approximate closed form solution for $p_B$. First, we assume that the buffers of individual servers are aggregated together. Then, the system can be modelled as a $M/M/N/NK$ system. However, such a model is more efficient than the actual system and thus under-estimates the blocking probability. Intuitively, we need to reduce the aggregated buffer size to reduce the amount of under-estimation. Since, on average, the number of tasks waiting in the server is $\frac{\overline{W}}{\mu}$, we postulate that the total buffer space available for sharing, $K_{eff}$, is given by $N(K - \frac{\overline{W}}{\mu}$. Therefore, we model the system as a $M/M/N/K_{eff}$ queue, and its blocking probability approximates $p_B$. Using the well established result of the blocking probability of a $M/M/N/k$ queue [18], $p_B$ is given by

$$p_B = \pi_N \left(\frac{A}{N}\right)^{K_{eff}-N}, \tag{10}$$

where

$$\pi_N = \left(E_N^{-1}(A) + \rho\frac{1-\rho^{K_{eff}-N}}{1-\rho}\right)^{-1}, \tag{11}$$

$A = N\lambda/\mu$, and $E_N(A)$ is the Erlang B blocking probability for a $M/M/N/N$ queue with offered traffic $A$.

 ISBN: 978-1-61208-711-5 

TABLE I. COMPARISON OF BLOCKING PROBABILITY OBTAINED BY $(p_K)^N$ AND SIMULATION

| | Buffer size | K = 5 | K = 10 | K =15 | K = 20 | K = 25 | K = 30 |
|---|---|---|---|---|---|---|---|
| $\lambda$ = 0.85 | $(p_K)^N$ | 4.58E-05 | 8.56E-08 | 6.77E-10 | 8.48E-12 | 1.27E-13 | 2.06E-15 |
| | simulation | 1.17E-02 | 6.13E-04 | 3.36E-05 | 8.79E-07 | 5.06E-08 | 5.89E-10 |
| $\lambda$ = 0.90 | $(p_K)^N$ | 1.76E-04 | 7.30E-07 | 1.56E-08 | 5.98E-10 | 3.05E-11 | 1.80E-12 |
| | simulation | 2.27E-02 | 3.27E-03 | 4.98E-04 | 7.12E-05 | 9.67E-06 | 1.31E-06 |

## V. NUMERICAL RESULTS

In this section, we use the developed analytical model to evaluate the performance and assess the potential benefits of the least loaded sharing scheme under various parameters. At the same time, we validate the analytical model by simulation. For this purpose, we have built a discrete event simulator in C++ to generate simulation results. We set $\mu = 1$ second, and vary the arrival rate $\lambda$ to obtain different loads. The duration of each simulation run varies according to the system parameters, ranging from $10^7$ to $10^{10}$ seconds, but the warm-up period is fixed at $10^5$ seconds.

First, we evaluate the blocking probability for $N = 5$, with various $K$ and $\lambda$. Table I compares the blocking probabilities obtained by $(p_K)^N$ and simulation, for K varying from 5 to 30 with a step of 5, with $\lambda = 0.85$ and 0.9, respectively. It shows that $(p_K)^N$ under-estimates the blocking probabilities by several order of magnitudes, and thus justifies the need of a more accurate way to calculate the blocking probabilities. Figure 4 and Figure 5 illustrate the effect of buffer size on the blocking probability for two different loads. It can be seen that, when $K$ increases, the blocking probability decreases exponentially. From the perspective of the validity of the proposed $M/M/N/K_{eff}$ model, the analytical results obtained by (10) are in good agreement with the simulation results. Furthermore, in comparison to the isolated scheme, the least loaded sharing scheme exhibits lower blocking probabilities. The reduction of blocking probabilities increases with buffer size. Clearly, this is because in the least loaded sharing scheme, the service and buffer capacity of all servers are aggregated together to serve the incoming tasks.

Figure 6 and Figure 7 depict the relationship between average waiting time experienced by tasks and the buffer size of each server in a fog computing system. It can be observed that the theoretical delay obtained by (9) is in good agreement with the results obtained from simulations. For both shared and isolated schemes, the mean delay of a task increases with the buffer size because more tasks are allowed to wait in the buffer. However, the delay incurred in the least loaded sharing scheme is always smaller than the isolated scheme. This is because, under the least loaded sharing scheme, tasks are more probable to enter a buffer with a shorter queue length.

It can be concluded that a single server working in isolation could reduce its blocking probability by simply increasing its buffer size. However, incoming tasks will suffer great system waiting time in such a system. On the other hand, the least loaded sharing scheme enables a fog computing system with heavy load and finite buffer size to offer low-latency processing of requests as well as low blocking probability.

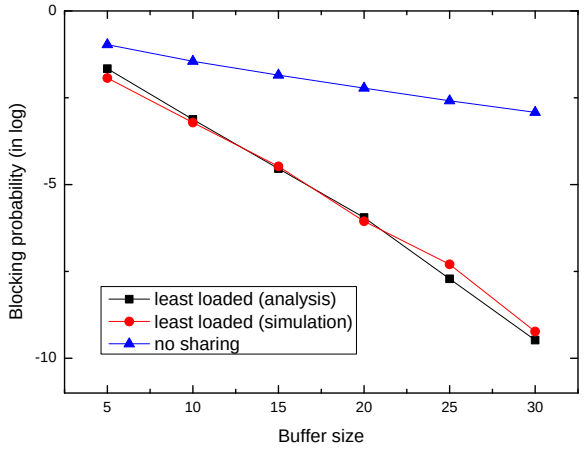



Figure 4. Blocking probability versus buffer size ($\lambda$ = 0.85).

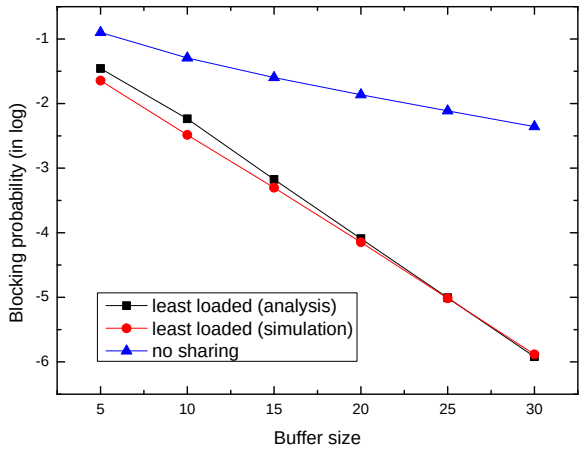



Figure 5. Blocking probability versus buffer size ($\lambda$ = 0.9).

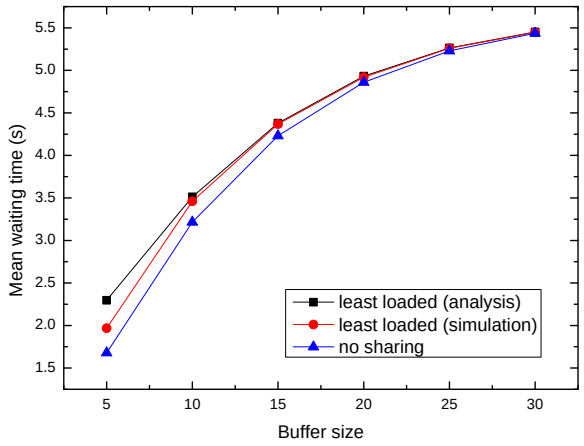



Figure 6. Mean delay versus buffer size ($\lambda$ = 0.85).

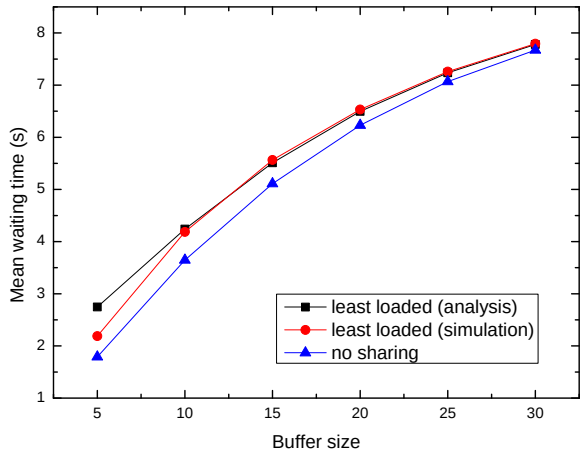



Figure 7. Mean delay versus buffer size ($\lambda$ = 0.9).

## VI. CONCLUSION

In this paper, we have proposed a least loaded sharing scheme for load balancing in a cluster of fog servers. We have developed an analytical model to evaluate the performance of the proposed scheme. The model is based on a state-dependent Markov chain. After solving the state probabilities of the Markov chain, the mean waiting time can be obtained. Also, a computationally efficient method has been developed to approximately calculate the blocking probability of requests. Simulation has been used to validate the model and show that the approximation is acceptable. Compared to the case when each server operates independently, our proposed scheme can utilize the resources of the cluster of fog servers more efficiently, leading to less waiting time and lower blocking probability experienced by users.

## REFERENCES

[1] A. Botta, W. D. Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey", *Future Generation Computer Systems*, vol. 16, no. 3, 2014, pp. 1391-1412.

[2] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", *Journal of internet services and applications*, 2010, vol. 1, no. 1, pp. 7-18.

[3] V. Sarathy, P. Narayan, and R. Mikkilineni, "Next generation cloud computing architecture: Enabling real-time dynamism for shared distributed physical infrastructure", *Proceedings, 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010*, pp. 48-53, 28-30 June, 2010, Greece.

[4] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. R. Choo, and M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework", *IEEE Access*, 2017, vol. 5, pp. 8284-8300.

[5] R. Kelly. "Internet of Things Data to Top 1.6 Zettabytesby 2022", [Online]. Available: https://campustechnology.com/articles/2015/04/15/internet-of-things-data-to-top-1-6-zettabytes-by-2020.aspx [retrieved: April, 2019].

[6] 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*, 2012, pp. 13-16.

[7] N. Peter, "Fog computing and its real time applications" *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, no. 6, pp. 266-269, 2015.

[8] X. Masip-Bruin, E. Marin-Tordera, G. Tashakor, A. Jukan, and G. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems", *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120-128, November 2016.

[9] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption", *IEEE Internet of Things Journal*, 2016, vol. 3, no. 6, pp. 1171-1181.

[10] R. Beraldi, A. Mtibaay, and H. Alnuweiri, "Cooperative Load Balancing Scheme for Edge Computing Resources", *Proceedings, 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC 2017)*, Valencia, Spain, 8-11 May, 2017, pp. 94-100.

[11] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing", *Proceedings, IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1-6.

[12] L. Gkatzikis and I. Koutsopoulos, "Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 24-32, June 2013.

[13] K. Kumar, J. Liu, Y. H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129-140, February 2013.

[14] W. Zhang *et al.*, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569-4581, September 2013.

[15] O. Munoz, A. Pascual-Iserte, and J. Vidal, "Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 10, pp. 4738-4755, October 2015.

[16] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo,"Joint allocation of computation and communication resources in multiuser mobile cloud computing," *Proceedings, 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp.26-30, 16-19 June 2013, Darmstadt, Germany.

[17] E. H. Elhafsi and M. Molle, "On the solution to QBD processes with finite state space", *Stochastic Analysis and Applications*, vol. 25, no. 4, pp. 763-779, 2007.

[18] M. Zukerman, *Introduction to Queueing Theory and Stochastic Teletraffic Models*, 2014. [Online]. Available: http://www.ee.cityu.edu.hk/zukerman/classnotes.pdf [retrieved: April, 2019].

 ISBN: 978-1-61208-711-5 

# A Scalable Architecture for Network Traffic Forensics

Viliam Letavay
Faculty of Information Technology
Brno University of Technology
Brno 61266, CZ
Email: iletavay@fit.vutbr.cz

Jan Pluskal
Faculty of Information Technology
Brno University of Technology
Brno 61266, CZ
Email: ipluskal@fit.vutbr.cz

Ondřej Ryšavý
Faculty of Information Technology
Brno University of Technology
Brno 61266, CZ
Email: rysavy@fit.vutbr.cz

***Abstract*—The availability of high-speed Internet enables new opportunities for various cybercrime activities. Security administrators and Law Enforcement Agency (LEA) officers call for powerful tools capable of providing network communication analysis of an enormous amount of network traffic as well as capable of analyzing an incomplete network data. Big data technologies were considered to implement tools for capturing, processing and storing packet traces representing network communication. Often, these systems are resource intensive requiring a significant amount of memory, computing power, and disk space. The presented paper describes a novel approach to real-time network traffic processing implemented in a distributed environment. The key difference to most existing systems is that the system is based on a light-weight actor model. The whole processing pipeline is represented in terms of actor nodes that can run in parallel. Also, the actor-model offers a solution that is highly configurable and scalable. The preliminary evaluation of a prototype implementation supports these general statements.**

***Keywords–Network forensic analysis; Network traffic processing; Actor model.***

## I. INTRODUCTION

The expansion of computer networks and Internet availability opens new opportunities for cybercrime activities and increases the number of security incidents associated with network applications. The number of connected devices grows, and traffic speed increases. Security administrators and Law Enforcement Agency (LEA) officers call for powerful tools that enable them to extract useful information from network communication [1]. The network forensics that is responsible for capturing, collecting and network data analyzing is becoming more important [2].

In the forensic investigation, the network traffic is continuously captured from multiple sources. The captured network data has a form of packet traces that have to be processed and analyzed up to the application layer. The network forensic tool has to decode protocols at different network layers of the Transmission Control Protocol/Internet Protocol (TCP/IP) model and various encapsulations. For LEA officers, interesting information lies in application messages, such as instant messaging, emails, voice, localizable information, documents, pictures, etc. The form and relevance of extracted artifacts may differ from case to case. Often, communication is encrypted. In this case, meta-data can be the only piece of information available. In all cases, the network forensic processing system has to be able to extract artifacts from the network traffic reliably, even if the packet capture is corrupted, for instance, some connections are incomplete, packets are malformed, or chunks of packets were not recorded because of capturing device issues.

The amount of data that needs to be processed to extract evidence from the network communication depends on the kind of a case that is investigated but usually gets large. It is very difficult to decode, extract and store the immense mass of information for further processing. We propose a distributed network forensic framework based on the *actor model* that is computation effective and capable of linear scalability. Scalable properties of *actor model* design for network forensics are promising, as shown by the Visibility Across Space and Time (VAST) platform [3]. Similarly to VAST, our solution provides real-time data ingestion and interactive data analysis, but in addition to VAST, we consider the full artifact extraction up to the application layer. Although it requires more computation resources, we demonstrate that it can still be achieved in a more straightforward and less resource consuming environment compared to Apache Hadoop technology, which is the norm for big data processing.

In Section II, we describe tools used by network forensics practitioners. Section III addresses issues faced by investigators and our proposed solution, which architecture is broadly discussed in Section IV. Section V evaluates preliminary performance results, and Section VI concludes the paper.

## II. BACKGROUND & RELATED WORK

Network forensics is a process that identifies, captures and analyzes network traffic. Network forensic techniques are used by several network forensic frameworks [4]–[9] and tools intended for intrusion detection (Zeek, VAST, Moloch) [10]–[12], network security monitoring (Microsoft Network Monitor, TShark, Wireshark, tcpdump) [13]–[16], and network forensic investigation for LEAs (Netfox Detective, PyFlag, NetworkMiner, EnCase, XPlico) [17]–[21]. Commonly available forensics tools are implemented either as a classic desktop or command line application or a traditional client-server solution.

To overcome the limitations of traditional tools, we propose to use distributed computing. The models for distributed processing [22][23] are more suitable for real-time network forensic analysis from multiple sources, such as logs and captured communication. The models are based on an agent system, where numerous agents perform the collection task. The extracted information is sent to the forensic network server

 ISBN: 978-1-61208-711-5 

and analyzed on this single node [24] only. The *forensic server* is the bottleneck that has to process all the data. To avoid this bottleneck, the Google Rapid Response (GRR) [25], a live forensic system, utilizes a cluster of servers. The system deploys agents running on users' computers that provide access to forensic information, e.g., remote raw disk and memory access. Processing of forensic data is done as flows. Each flow is maintained on the server. Server nodes run workers that process the active flows. Adding more server nodes enables to run more workers and thus it is possible to handle more clients simultaneously.

Elimination of bottlenecks in the architecture offers scalability and improved reliability. The *actor model* [26] is one of the attractive solutions that address the problem elegantly and efficiently. It comes with a separate unit called an *actor*. Actors execute independently and in parallel. They communicate with each other asynchronously via message passing, and their state is otherwise immutable. Actors are capable of spawning new actors, forming a parent-child relationship, allowing the creation of a tree-like structure of actors. Actor's current behavior determines how it processes the incoming messages. Every actor in an actor system is uniquely identified by an address which other actors use as destinations of the messages they want to send out. This address can identify actors at the local machine and also the ones at the remote machines, allowing easy means of communication between nodes of a cluster. Compared to another similar programming model, the *Communicating Sequential Processes* (CSP) [27], elementary units of computation – processes are anonymous and communicate with each other via established communication channels. The actor system is the key enabler for the VAST system [3]. In VAST, actors implement importing, archiving, indexing and exporting processed data. Actors live in nodes that map to system processes. The system scales by creating more nodes either on the single machine or a cluster of computers.

Moloch is another tool, worth to mention, that uses principles of distributed computing for massive scale network traffic monitoring, full packet capturing and indexing [12]. Moloch system consists of sensors that capture the communication and Elasticsearch database that is a distributed search and analytics engine. The system scales by adding new nodes running Elasticsearch instances.

## III. Problem Statement and Solution

Our goal is to design and create a system capable of long-term, high-speed, real-time network traffic filtering and processing up to the application layer. The software solution should be scalable and hardware independent. To achieve this, we have to deal with the challenges elaborated in the rest of this section.

### A. Architectural Design

*How to create a system for packet filtering and analysis of communication that can identify application protocols, gets forensics artifacts and searches through them?*

Network forensics is a tedious work that strictly relies on completeness and precision of all undertaken steps to gain a piece of a puzzle that fits together as a shred of evidence. Considering the current speeds of regular users' home network connection(s), a comprehensive classical analysis on a single machine would require enormous computation resources. Try to imagine, that each network packet would be analyzed by many protocol dissectors with a goal to extract, for example, an acknowledgment of email delivery. To achieve this goal, with optimal computational resources, we must revisit currently utilized methods and redesign them to work in a distributed environment which brings new challenges to architecture design, application of algorithms, data synchronization, and so on.

### B. Scalability on Commodity Hardware

*How can the solution be scalable and hardware independent despite the hardware limitations?*

Let us consider this imaginary demonstration. The math is simple, one computer with $1\,\mathrm{Gbps}$ Network Interface Card (NIC) that has a relatively simple task to capture traffic during full line load would be required to write to a disk under the constant speed of $1000\mathrm{Mbps} \approx 125\,\mathrm{MB/s}$. Our system has to guarantee that no data loss occurs during the capture. A suspect can simultaneously download and upload data which means that the monitoring device cannot have only one $1 * 1\,\mathrm{Gbps}$ NIC, but it needs $2 * 1\,\mathrm{Gbps}$ cards, one for uplink, one for downlink. Thus, the required speed of continuous disk writing would be $2 * 125\,\mathrm{MB/s} \approx 250\,\mathrm{MB/s}$. Now, if the requirement is to store the communication for one day, the disk capacity has to be $250\,\mathrm{MB/s} * 86\,400\,\mathrm{s} \approx 21.6\,\mathrm{TB}$. This is achievable with commodity hardware, e.g., $2 * 12\,\mathrm{TB}$ drives with Redundant Array of Inexpensive Disks (RAID) 0 or $4 * 12\,\mathrm{TB}$ with RAID 1+0 — assuming higher write/read speed than $250\,\mathrm{MB/s}$. However, what if only one day is not enough? For a typical forensic case, capturing period spawns through weeks or months.

From our previous experiments, we know that a single computation node is limited and commodity hardware is hardly sufficient to perform all required operations in real-time and over long periods. Separation of frames into a conversation which requires a dissection of the network protocols up to the application layer, which speed is roughly $300\,\mathrm{Mbps}$ [28, pp. 45-51] is not sufficient. On the other hand, we are confident that the application created and optimized for this singular purpose can do the processing faster and breach the $1\,\mathrm{Gbps}$ line speed. Nevertheless, we do not believe that a single machine solution with commodity hardware is capable of doing overall analysis and extraction of information from the application layer. We have to design our solution as a distributed system across multiple machines.

### C. Overall Performance

*What scalability and acceleration of data processing can be achieved?*

The proposed solution is based on the actor model. Each actor represents an independent processing unit. The communication between actors is managed by messaging. Actors have no shared state; thus all of them can work in parallel. If actors run on the same node, the message passing has little additional overhead compared to a function call or a loop. However, if actors scale over multiple nodes, messages need to be serialized. This process introduces latency and consumes part of the processing power. The scalability of the actor model is linear [3].

 ISBN: 978-1-61208-711-5 

## IV. Architectural Design

Incomplete data provided by unreliable traffic interception can lead to inaccurate results; some information may be lost, some fabricated by reconstruction process [29]. Keeping the above facts in mind, the processing cannot strictly follow Requests for Comments (RFCs) and behave like a *kernel* network stack implementation, but it has to incorporate several heuristics. For example, to fill missing gaps in data, and to consider these fillings during application protocol processing, or never to join multiple frames into a single conversation unless it passes more advanced heuristic-based checks. Network forensic tools that we have worked with do mostly respect RFCs and thus may produce misleading results, as shown by Matousek et al. [29].

We propose a distributed architecture composed of commodity hardware that will be capable of linear scalability, and capable of efficient resource utilization. The overall architecture is shown in Figure 1.

At the top level, we have divided the entire process into the two main stages:

- Data preprocessing — The reconstruction of conversations at the application layer (L7) of the TCP/IP model. This process consists of consecutive segregation of captured communication into the internet (L3) and transport (L4) conversations and deploying a reassembling heuristics [29] to recognize individual L7 conversations inside a parent L4 conversations and to reassemble their payloads with respect to data loss, reordering or duplication. Every L7 conversation holds information about the source and destination endpoints (IP addresses, ports), timestamps, type of transport protocol (UDP or TCP) and reassembled payloads of exchanged application messages.
- Data analysis — The analysis of each application conversation consists of the identification of the application protocol, and extraction of application events, e.g., visited web pages, exchanged emails, domain name queries, etc., with proper application protocol dissector that yields sets of forensic artifacts.

### A. Data Prepossessing

*The First stage* is executed on a set of independent *Reassembler* nodes. These reconstruct L7 conversations from the stream of captured packets which can originate from *Packet Capture (PCAP) files* or can be captured from *the live network interface*.

In the most common use-case, we have one source stream (i.e., one PCAP file) which we want to analyze. Therefore, to utilize multiple *Reassembler* instances, we have to split packets from this stream into smaller sub-streams, which will be distributed among available *Reassembler* instances. For this split, we cannot use a naive method such as *Round Robin*, because *Reassembler* nodes operate independently of each other and to fully reconstruct L7 conversation a particular *Reassembler* has to obtain all the pieces of that particular L7 conversation. In case we would use *Round Robin*, a situation could occur when half the packets from one L7 conversation would end up in one *Reassembler* node and the second half in another; both nodes would have incomplete data and none of them would be able to reconstruct the conversation entirely.

Our proposed solution to this problem is another type of node – *L4 Load Balancer*, which will be positioned in front of the *Reassembler* nodes and which, as a name suggests, distributes packets based on their associations to L4 conversations each of which can consist of multiple L7 conversations. *L4 Load Balancer* extracts source and destination IP addresses and ports and transport protocol from each packet of the source stream and uses this information to decide to which instance from the available *Reassemblers* should it forward to. This way, all packets of a particular L7 conversation will always be forwarded to only one *Reassembler* instance.

*Reassemblers* build a tree-like structure of L3 and L4 conversations which are represented by the actors. Each received packet is first forwarded to an appropriate L3 conversation actor, which in turn forwards it further down to an appropriate L4 conversation actor which reassembles L7 conversations. This segregation of packets into the individual L4 conversations before actual L7 conversation reassembling is required, as implemented reassembling heuristics expect to operate on packets from a single L4 conversation at the time. The use of a hierarchical actor design allows us to perform independent portions of the processing in parallel and also to easily implement management strategies such as passing management messages to a particular L3 conversation actor and its children L4 conversation actors. The reconstructed L7 conversations are stored in a distributed database, ready to be retrieved in the second stage of the execution.

### B. Data Analysis

In the *second stage*, a subset of reconstructed L7 conversations is retrieved from the distributed database and delivered to the *Application protocol dissector* nodes. For every L7 conversation, *Application protocol dissector* nodes identify the used application protocol and use a proper dissector module dedicated to the processing of a single application protocol, such as Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) or Domain Name System (DNS), to extract application protocol messages from this L7 conversation. Obtained data are stored back into the distributed database. Processing of application messages is under normal circumstances possible only with unencrypted network communication. From Secure Sockets Layer/Transport Layer Security (SSL/TLS) communication which encapsulates application protocols, such as HTTP, we can extract only unencrypted portions of this data such as the server's cryptographic certificate. Possible ways to decrypt and subsequently, parse the SSL/TLS communication is to own a private key of a given SSL/TLS server or to deploy an SSL/TLS intercepting proxy [30].

## V. Preliminary Evaluation

Our prototype implementation is based on C# actor system library *Akka.NET*. For testing and performance benchmarking, we have implemented two modes of operation:

1) *Offline* — isolated execution which combines the functionality of a single *L4 Load Balancer* and *Reassembler* node inside a single system's process. No inter-actor message serialization is therefore required.
2) *Online* — distributed execution spanning across multiple cluster nodes. The inter-actor message serialization is required as messages destined to remote

 ISBN: 978-1-61208-711-5 

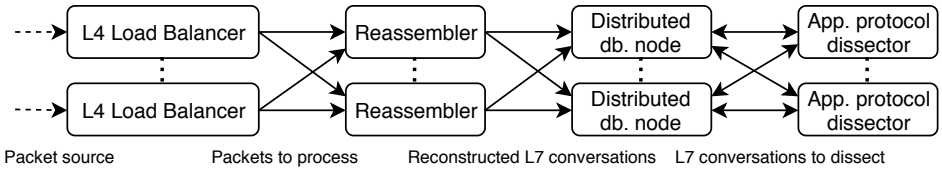



Figure 1: Architecture diagram showing the proposed system nodes with information flow between them.

actors (nodes) have to leave an originating system's process and be transmitted over a computer network in a serialized format. This introduces additional latency and performance overhead.

Additionally, for proof-of-concept benchmarking, the functionality of *Application protocol dissector* nodes was included inside *Reassembler* nodes to eliminate distributed database as a middleman between them. In the following measurements, we focus on a raw network capture's processing performance of the so-far naive implementation. Currently, our prototype implementation supports the dissection of two application protocols (DNS and HTTP).

We have measured the preliminary performance of the implementation on two different hardware configurations:

- *Workstation* — Intel i7-5930K 4.3 GHz, 12 cores, 64 GB RAM, 512 GB SSD
- *Mini-cluster* — 4x servers with Intel Xeon E5520, 2.26 GHz, 8 cores, 48 GB RAM, 1 TB SSD, 1 Gbps network

We used a public data set of M57-Patents Scenario [31], that consists of real-world data captured over a month. We merged all network traces into one PCAP file of roughly 4.8 GB and 5,707,845 frames. One large PCAP file simulates our use-case of streamed-in communication that needs to be load-balanced from a single node.

We started with measurements in an *offline* mode on a single machine, firstly with a PCAP file parsing operation and incrementally added consequent operations and measured processing speeds, as Table 1 describes. Preliminary evaluation suggests that the *raw speed* of roughly 3.8 Gbps, for PCAP file reading and packet parsing is sufficient. The process of reconstructing L7 conversations that segregates IP flows by packet source and destination IP addresses, ports and transport protocol type with additional heuristics [29], that also reassembles TCP/UDP streams, is computationally heavier, reaching "only" 942 Mbps, and is about 4x slower than only read and parsing. With added HTTP & DNS dissection, performance slightly decreased further down to 880 Mbps.

TABLE 1. PROCESSING SPEEDS OF OUR OFFLINE TEST SCENARIO ON A SINGLE MACHINE

| | **Workstation** [Mbps] | **Mini-cluster node** [Mbps] |
|---|---|---|
| **PCAP file reading** | 5103 | 5719 |
| **Packet parsing** | 3853 | 1679 |
| **L7 Conversation tracking** | 942 | 380 |
| **HTTP & DNS extraction** | 880 | 358 |

The *CPU frequency* (performance per CPU core) plays a very important part in overall performance, that can be observed if we compare our *Workstation* with node from *Mini-cluster* — 880 Mbps vs. 358 Mbps. All other components except CPUs are otherwise roughly comparable as we can see by comparing the speed of "PCAP file reading".

The scalability is described in Table 2 that shows performance in *online* mode. The solution was deployed on *Mini-cluster*. The first node was reading the captured communication from a PCAP file and load-balancing it to the rest that reassembled L7 conversations and extracted HTTP and DNS artifacts. In the measurements, we can see an increase in the performance with each added *Reassembler*. When compared with the results in Table 1, the performance of a distributed processing at the *Mini-cluster* exceeded that of a single node running in an *offline* mode. Nevertheless, further optimization is required to achieve linear scalability as a single *L4 Load Balancer* fails to fully saturate available *Reassemblers* by distributing the packets fast enough. We have observed that serialization of messages containing the packets to process heavily contributes to the overall computational complexity and easily becomes a bottleneck of our solution.

TABLE 2. PROCESSING SPEEDS OF OUR ONLINE TEST SCENARIO MEASURED ON MINI-CLUSTER

| **Reassemblers count** | **One** [Mbps] | **Two** [Mbps] | **Three** [Mbps] |
|---|---|---|---|
| **HTTP & DNS extraction** | 233 | 407 | 453 |

We compare our solution, called Network Traffic Processing & Analysis Cluster (NTPAC), running in the *offline* mode at the *Workstation* with commonly used network forensic tools in Table 3. Our solution is an order of magnitude faster while delivering a comparable amount of results in terms of reconstructing L7 conversations and extracting HTTP and DNS artifacts.

TABLE 3. PROCESSING SPEEDS OF COMMONLY USED NETWORK FORENSIC TOOLS MEASURED ON WORKSTATION

| **NTPAC** [Mbps] | **Netfox** [Mbps] | **Wireshark** [Mbps] | **NetworkMiner** [Mbps] |
|---|---|---|---|
| 880 | 65.6 | 73.4 | 15.8 |

## VI. CONCLUSION

In this research, we proposed a system for distributed real-time forensic network traffic analysis up to the application layer capable of large-scale communication processing. We intend to create a system based on the actor model that scales linearly and is hardware independent. The implementation environment of the .NET Core framework and C# language enables rapid development compared to C/C++ that is used by VAST and Moloch. Also, our solution is multiplatform and easily staged with Docker Swarm. Therefore, the deployment of the entire distributed application at the computation

 ISBN: 978-1-61208-711-5

cluster is reduced to one command. The solution is distributed under the MIT License and hosted as an open-source project on GitHub here [32].

In the near future, we plan to measure the performance of our solution using data from real-world cases. Because of legal reasons, deployment to public cloud infrastructure is out of the question. Therefore, we need to build a private one that consists of nodes with high CPU frequencies and 10 Gbps network interfaces. Additionally, we need to profile and optimize processing and distribution mechanisms, to expand the set of protocols supported by application protocol dissectors and to add support for tunneling mechanisms.

## VII. Acknowledgement

This work was supported by BUT project "ICT Tools, Methods and Technologies for Smart Cities" (2017-2019), no. FIT-S-17-3964.

## References

[1] N. Beebe, "Digital forensic research: The good, the bad and the unaddressed," in IFIP International Conference on Digital Forensics. Springer, 2009, pp. 17–36.

[2] E. S. Pilli, R. C. Joshi, and R. Niyogi, "Network forensic frameworks: Survey and research challenges," digital investigation, vol. 7, no. 1-2, 2010, pp. 14–27.

[3] M. Vallentin, "Scalable network forensics," Ph.D. dissertation, UC Berkeley, 2016.

[4] S. Rekhis, J. Krichene, and N. Boudriga, "Digfornet: digital forensic in networking," in IFIP International Information Security Conference. Springer, 2008, pp. 637–651.

[5] A. Almulhem and I. Traore, "Experience with engineering a network forensics system," in International Conference on Information Networking. Springer, 2005, pp. 62–71.

[6] W. Wang and T. E. Daniels, "A graph based approach toward network forensics analysis," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 1, Oct. 2008, pp. 4:1–4:33.

[7] N. L. Beebe and J. G. Clark, "A hierarchical, objectives-based framework for the digital investigations process," Digital Investigation, vol. 2, no. 2, 2005, pp. 147–167.

[8] S. Perumal, "Digital forensic model based on malaysian investigation process," International Journal of Computer Science and Network Security, vol. 9, no. 8, 2009, pp. 38–44.

[9] W. Halboob, R. Mahmod, M. Abulaish, H. Abbas, and K. Saleem, "Data warehousing based computer forensics investigation framework," in 2015 12th International Conference on Information Technology-New Generations (ITNG). IEEE, 2015, pp. 163–168.

[10] Zeek, [retrieved: April, 2019]. [Online]. Available: https://www.zeek.org/

[11] Vast, [retrieved: April, 2019]. [Online]. Available: http://vast.io/

[12] Moloch, [retrieved: April, 2019]. [Online]. Available: https://molo.ch/

[13] Microsoft Network Monitor, [retrieved: April, 2019]. [Online]. Available: https://support.microsoft.com/en-us/help/933741/information-about-network-monitor-3

[14] TShark, [retrieved: April, 2019]. [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html

[15] Wireshark, [retrieved: April, 2019]. [Online]. Available: https://www.wireshark.org/

[16] TCPDUMP, [retrieved: April, 2019]. [Online]. Available: https://www.tcpdump.org/

[17] Netfox Detective, [retrieved: April, 2019]. [Online]. Available: https://github.com/nesfit/NetfoxDetective

[18] PyFlag, [retrieved: April, 2019]. [Online]. Available: https://github.com/py4n6/pyflag

[19] NetworkMiner, [retrieved: April, 2019], https://www.netresec.com/?page=NetworkMiner.

[20] EnCase, [retrieved: April, 2019]. [Online]. Available: https://www.guidancesoftware.com/encase-forensic

[21] XPlico, [retrieved: April, 2019]. [Online]. Available: https://www.xplico.org/

[22] W. Ren and H. Jin, "Distributed agent-based real time network intrusion forensics system architecture design," in Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on, vol. 1. IEEE, 2005, pp. 177–182.

[23] D. Wang, T. Li, S. Liu, J. Zhang, and C. Liu, "Dynamical network forensics based on immune agent," in Natural Computation, 2007. ICNC 2007. Third International Conference on, vol. 3. IEEE, 2007, pp. 651–656.

[24] S. Khan, A. Gani, A. W. A. Wahab, M. Shiraz, and I. Ahmad, "Network forensics: Review, taxonomy, and open challenges," Journal of Network and Computer Applications, vol. 66, 2016, pp. 214–235.

[25] M. Cohen, D. Bilby, and G. Caronni, "Distributed forensics and incident response in the enterprise," Digital Investigation, vol. 8, 2011, pp. S101 – S110, the Proceedings of the Eleventh Annual DFRWS Conference. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287611000363

[26] C. Hewitt, P. Bishop, and R. Steiger, "A universal modular actor formalism for artificial intelligence," in Proceedings of the 3rd International Joint Conference on Artificial Intelligence, ser. IJCAI'73. Morgan Kaufmann Publishers Inc., 1973, pp. 235–245.

[27] C. A. R. Hoare, "Communicating sequential processes," Commun. ACM, vol. 21, no. 8, Aug. 1978, pp. 666–677.

[28] J. Pluskal, "Framework for captured network communication processing," Master's thesis, FIT BUT, 2014.

[29] P. Matoušek et al., "Advanced techniques for reconstruction of incomplete network data," in International Conference on Digital Forensics and Cyber Crime. Springer, 2015, pp. 69–84.

[30] S. Davidoff and J. Ham, Network Forensics: Tracking Hackers through Cyberspace. Prentice Hall, 2012.

[31] M57-Patents Scenario, [retrieved: April, 2019]. [Online]. Available: https://digitalcorpora.org/corpora/scenarios/m57-patents-scenario

[32] NTPAC, [retrieved: April, 2019]. [Online]. Available: https://github.com/nesfit/NTPAC/

 ISBN: 978-1-61208-711-5 

# Network Diagnostics Using Passive Network Monitoring and Packet Analysis

Martin Holkovič

CESNET a.l.e.
Zikova 1903/4
Prague 16000, CZ
Email: holkovic@cesnet.cz

Ondřej Ryšavý

Faculty of Information Technology
Brno University of Technology
Brno 61266, CZ
Email: rysavy@fit.vutbr.cz

***Abstract*—Finding a problem cause in network infrastructure is a complex task because a fault node may impair seemingly independent components. On the other hand, most communication protocols have built-in error detection mechanisms. In this paper, we propose to build a system that automatically diagnoses network services and applications by inspecting the network communication automatically. We model the diagnostic problem using a fault tree method and generate a set of rules that identify the symptoms and link them with possible causes. The administrators can extend these rules based on their experiences and the network configuration to automatize their routine tasks. We successfully deployed the proof-of-concept tool and found interesting future research topics.**

***Keywords–Network diagnostics; passive network monitoring; rule-based diagnostics; fault tree analysis; event-based diagnostics.***

## I. Introduction

Network infrastructure and applications are complex, prone to cyber attacks, outages, performance problems, misconfiguration errors, and problems caused by software or hardware incompatibility. All these problems may affect network performance and user experience [1] which may cause fatal problems in critical networks (e.g., E-health, Vanet, Industrial IoT).

Many network administrators do not have the proper tools or knowledge to diagnose and fix network problems effectively, and they require an automated tool to diagnose these errors [2]. Zeng et al. [3] provide a short survey on network troubleshooting from the administrators' viewpoint identifying the most common network problems: *reachability problems, degraded throughput, high latency, and intermittent connectivity*. The consulted network administrators expressed the need for a network monitoring tool that would be able to identify such problems.

This paper proposes a system which creates diagnostic information only by performing passive network traffic monitoring and packet-level analysis. Previous research and development provided tools for helping administrators to diagnose faults [4] and performance problems [5]. However, these tools either require *installation of agents on hosts, active monitoring, or providing rich information about the environment*.

One of the most common ways of analyzing network traffic is by using a network packet analyzer (e.g., Wireshark). The analyzer works with captured network traffic (PCAP files) and displays structured information of layered protocols contained in every packet (encapsulated protocols, protocol fields). Administrators work with this information, check transferred content and compare the data with expected values. This process, done manually, is time-consuming and requires a good knowledge of network protocols and technologies.

The main contribution of this paper is a proposal of a tool for automatic diagnoses of network related problems from network communication only. Our approach tries to imitate a diagnostic process of a real administrator using the fault tree method and a popular packet parsing tool tshark. We have also implemented a proof-of-concept implementation to confirm the viability of the approach.

The paper is organized as follows. Section 2 defines the problem statement and research questions. Section 3 discusses related work and describes diagnostic approaches. Our solution consists of three stages and is introduced in Section 4. Section 5 instructs network administrators how to use our system (proof-of-concept) and shows how we model diagnostic knowledge. Finally, Section 6 is the conclusion which summarizes the current state and proposes future works.

## II. Research Questions

Our primary goal is to design a system that infers possible causes accountable for network related problems, such as service unreachability or application errors. Offering a list of actions for fixing the errors' cause is the secondary and optional goal. All this information is gathered only from captured network communication.

In our work, we focus on enterprise networks that have complex networking topologies, usually consisting of heterogeneous devices. We expect that the administrators will collect network communication on appropriate places and validate its consistency before the analysis.

To achieve our goal, we need to find answers to the following research questions:

1) How to model different network faults in a suitable way for implementation in a diagnostic system? *Reachability, application specific, and device malfunctioning problems* can cause various networking issues. We need to have a unified approach for modeling these problems to identify the symptoms and link them with root causes.
2) What information should be extracted from the captured network communication to identify symptoms of failures?

 ISBN: 978-1-61208-711-5 

In our case, we can passively access the communication in the monitored network and extract the necessary data to detect possible symptoms. An approach that can efficiently detect the symptoms in terms of precision and performance is needed.

3) How to identify the root cause of the problem, if we have a set of identified symptoms? The core part of the diagnostic engine is to apply knowledge gathered from observed symptoms to infer the possible root cause of the observed problem. The result should provide the information in sufficient detail. For instance, if the process on server crashed, then we would like to know this specific information instead of a more general explanation (e.g., a host failure has occurred).
4) What list of actions can we give to the administrator to fix the problems? Based on the observed symptoms and the root cause, the system should be able to provide fixing guidelines. These guidelines are supposed to be easy to understand even for an inexperienced administrator.

## III. Related Work

A lot of research activities were dedicated to the diagnoses of network faults. Various methods were proposed for different network environments [4], in particular, home networks [6], enterprise networks [7]–[10], data centers [5], backbone and telecommunications networks [11], mobile networks [12], Internet of Things [13], Internet routing [14] and host reachability. Methods of network troubleshooting can be roughly divided into the following classes:

**Active methods** use traffic generators to send probe packets that can detect the availability of services or check the status of applications [15]. Usually, generators create diagnostic communication according to the test plan [7]. The responses are evaluated and provide diagnostic information that may help to reveal device misconfiguration or transient fail network states. Diagnostic probes introduce extra traffic, which may pose a problem for large installations [10]. Also, active methods may rely on the deployment of an agent within the environment to get information about the individual nodes [8].

**Passive methods** detect symptoms from existing data sources, e.g., traffic metadata [11], traffic capture files, network log files [14], performance counters. Passive methods can utilize the data provided by network monitoring systems.

Of course, the proposed systems also combine passive traffic monitoring to detect faults with active probing to determine the cause of failure. Identifying anomalies related to network faults and linking them with possible causes can be done by using one of the following approaches:

**Inference-based** approach uses a *model* to identify the dependence among components and to infer the faults using a collection of facts about the individual components [8], [16].

**Rule-based** approach uses *predefined rules* to diagnose faults [9]. The rules identify symptoms and determine how these contribute to the cause. The rules may be organized in a collaborative environment for sharing knowledge between administrators [6].

**Classifier-based** approach *requires training data* to learn the normal and faulty states. The classifier can identify a fault and its likely cause [17].

Network diagnostics based on traffic analysis can also use methods proposed for anomaly detection as some types of faults result in network communication anomalies.

Main contributions of our solution:

- automation of the tool Wireshark - Wireshark is a well-known protocol analyzer but lacks any task automation;
- the result is well understandable - the result contains steps which a real administrator would execute;
- easily extendable list of rules - the rules use Wireshark display filter language [18].

## IV. Proposed System Architecture

We have built a proof-of-concept expert system to analyze network traffic. The system combines rule-based and inference-based approaches. We will not use a classifier-based approach [19], because it requires too much training data and only returns the root cause of the problem and not how it relates to the detected symptoms. Another benefit of the rule-based approach is that we can cover very specific situations for which getting training data could be very problematic.

We are focusing purely on passive methods because active methods are generating additional traffic into diagnosed networks (which is not acceptable for us) and also because this way, an administrator can perform an offline analysis on a computer not connected to the diagnosed network.

The proposed system processes the input data in several stages as shown in Figure 1. The first stage labeled as *Protocols Analyzer* filters and decodes input packets using an external tool. The second stage named *Events Finder* executes simple rules to identify events significant from the diagnostics point of view. In the third stage (*Tree Engine*), decision trees identify the possible problem cause and create a diagnostic output. All stages are easily extendable by the administrator who can add new rules and definitions.

Our proposed approach can also use different data sources (e.g., log files) as shown in Figure 1. *Events Finder* searches through data using analyzers specific to each data source. Additional analyzers could increase the diagnostic capability, however in our research, we are focusing only on network data, and we leave other possibilities for future research.

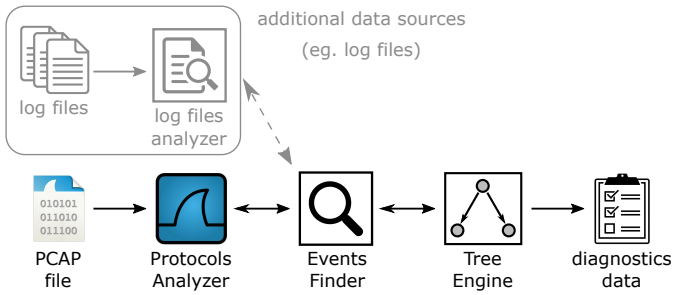



Figure 1. Top level architecture design of all the proposed system stages. The gray area represents optional extensions — additional data sources.

### A. Protocols Analyzer

The first step in the processing pipeline is decoding captured network traffic in the PCAP format into a readable JSON format. We employ the tool *tshark*, which is a command

line version of the widely-used network protocol analyzer Wireshark. Because tshark follows the field naming convention used by Wireshark, we can use Wireshark Display Filter Expressions to select packet attributes. Tshark supports all packet dissectors available in Wireshark. Using tshark brings the following benefits:

- huge support of network protocols and when a new protocol is created, community can implement parsers very quickly and for free;
- adds tunneled, segmented and reassembled data support;
- tshark marks extracted data with the same names as displayed inside the Wireshark GUI. This allows a creation of easy-to-read API for diagnostics;

### B. Events Finder

Events finder aims to identify events useful for network diagnosis (for example, a successful SMTP authentication event). An event rule consists of two parts: a list of packet filters and a list of assertions to express additional constraints. Both filter expressions and assertions use Wireshark's display filter language. Using this language, the expressions can be first tested in Wireshark before we use them in event finder rules.

The system evaluates the event rule as follows: (i) Each packet filter returns a list of packets matching the filter. (ii) Assertions are evaluated to select pairs of packets satisfying the constraints. A result has the form of a collection of pairs of packets, e.g., a rule that identifies DNS request-response pairs asserts that the transaction ID in both the request and response packets match. Assertion expressions use the display filter language extended with basic mathematical operations.

The event rules have the declarative specification written in YAML format. This format is described in subsection V-B. Rules are organized into modules. New modules can be easily added extending the rule database.

### C. Tree Engine

The tree engine infers the possible error cause by evaluating a decision tree that contains expert knowledge about supported network protocols and services. Each node of the tree contains a diagnostic question. Questions refer to events identified by the *Event Finder*. Paths in the tree represent gathered knowledge and lead to the possible cause of the problem. Along the path, a diagnostic report is created to provide additional information for experienced users. The diagnostic report is produced in a human-readable format, as well as in a machine format useful for further processing or visualization.

The decision tree is comprised of the declarative specification of tree nodes enriched by Python code. Injection of Python code into the tree node definitions enables us to do complex knowledge processing. The idea is to keep the declarative part simple enough for most of the use-cases. The Python code is needed for specific use-cases, where a custom processing logic is necessary. The tree is defined using the YAML format rules, and subsection V-A describe its syntax.

## V. Rule Specification

Diagnostic engine defines each protocol as a decision tree. The tree consists of nodes representing administrator questions, and edges representing answers to these questions. The edge

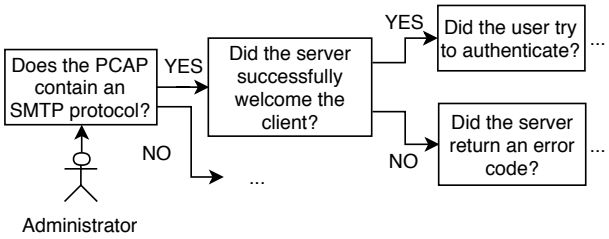



Figure 2. A simple illustration of a binary decision tree. Administrator diagnoses SMTP problem by checking questions in the predefined order.

can move the diagnostic process from one question to another or finish the process with the discovered result.

The questions simulate thinking of a real administrator. Typically, an administrator starts to search for certain network packet values and after the search for them is finished, the administrator searches for next values based on the result. In our solution, each question can only have two answers: success or fail. This yields a binary decision tree. Figure 2 shows an example of a small portion of the SMTP tree.

We need to convert the decision tree to a format understandable by our system. This conversion is split into two steps: 1) defining tree nodes (*Tree node rules*) and 2) defining conditions for choosing tree nodes (*Event condition rules*). The following subsection describes the syntax for both tree node rule and event condition rule. The reason why a node rule does not contain a condition code directly is that multiple rules would not be able to use the same condition code (reusability).

Conversion assigns a name to each node (*label_id*). We use the node names as labels for switching from one node to another. Each node has a condition (*condition_rule_id*), defined as an *Event condition rule*, used for choosing the next diagnostic step. Each rule can have one or none success and fail branch (*branch_code*). Branches contain executable Python code and the next node rule name. After the execution of the Python code, the analysis switches to the next node. Figure 3 shows the pseudocode for writing tree nodes.

```
1 label_id:
2   if (condition_rule_id):
3     success branch_code
4   else:
5     fail branch_code
```

Figure 3. Pseudocode for writing a tree node. Each node should have a unique id, condition, and branch codes.

### A. Tree Node Rules

Each rule consists of an *event condition rule* name which should be executed, next states and blocks of Python code. The Python code can process packet data, make logical decisions and most importantly, generate diagnostic output. Instead of writing the whole output inside these rules, the rule contains only the name of the event. Each rule can switch to another protocol rule to diagnose problems across several protocols, e.g., if an SMTP communication is not detected, we will check if there are any ICMP unreachable messages, failed TCP connection attempts or incorrect DNS resolutions. Figure 4 shows an example of one rule defying the middle node from the tree in Figure 2.

 ISBN: 978-1-61208-711-5

```
id: smtp detected # name of the rule
query: welcome ok? # Events Finder rule
success:
  state: client welcomed # next state
  code: | # Python code follows
    event("client_welcomed")
fail:
  state: check error # next state
  code: | # Python code follows
    event("client_not_welcomed")
```

Figure 4. Simple Tree Engine rule showing what should be done if SMTP server welcomed the client or not.

### B. Event Condition Rules

Rules in this section describe *how* the question is converted into packet lookup functions. Each rule may look for several independent packets, which are combined and checked if their relation fulfills the assert condition. Each question returns a list of tuples, where a tuple represents packets fulfilling the assert condition. Figure 5 shows an example of a simple rule for the question *Did the server successfully welcome the client?* Section *facts* looks for any hello commands and OK responses. The system puts founded packets which belong together into touples based on the *asserts* section.

```
id: welcome ok? # name of the rule
facts: # which packets we are looking for
  command: smtp.req.command in {"HELO"
           "EHLO"}
  reply: smtp.response.code == "250"
asserts: # packets relation constrain
 -command[tcp.stream]==reply[tcp.stream]
 -command[tcp.ack]==reply[tcp.seq]
```

Figure 5. Example of SMTP rule for checking if the server welcomed the client or not.

SMTP: Connection detected
SMTP: Server welcomed the client
SMTP: Server is ready
SMTP: Authentication 'gurpartap@patriots.in' - ok
SMTP: The communication is not encrypted
SMTP: No email has been sent
SMTP: Transaction error code 552 - Requested mail actions aborted - Exceeded storage allocation
SMTP: Empty email account storage (check SPAM folder) or increase the account quota.

Figure 6. An example of diagnostic output for an SMTP error. After an error 552 is detected and translated into human-readable error description, the system proposes a list of actions for fixing the error.

Before executing the diagnostic process, it is necessary to define event names from the *Tree engine* rules. A definition is just a simple dictionary which contains a severity and a description message. Part of the event description can be a pointer to another dictionary, which translates error codes to a human readable format. For example, instead of SMTP error code 552, the message "Requested mail actions aborted - Exceeded storage allocation" is displayed. After all rules and events are defined, it is possible to execute the diagnostic process. Figure 6 shows an example of one diagnostic output.

Table 1. Supported protocols and amount of rules and success, warning, error events which describe various protocol behavior situations.

| Protocol | Node rules | Condition rules | Events | | |
|---|---|---|---|---|---|
| | | | Success | Warning | Error |
| **DHCP** | 25 | 23 | 10 | 9 | 4 |
| **DNS** | 12 | 12 | 8 | 2 | 6 |
| **FTP** | 24 | 10 | 17 | 5 | 6 |
| **ICMP** | 4 | 2 | 0 | 0 | 4 |
| **IMAP** | 15 | 8 | 7 | 0 | 11 |
| **POP** | 21 | 7 | 8 | 5 | 10 |
| **SIP** | 38 | 22 | 15 | 1 | 8 |
| **SLAAC** | 8 | 7 | 1 | 5 | 2 |
| **SMB** | 27 | 25 | 20 | 4 | 5 |
| **SMTP** | 17 | 13 | 10 | 5 | 9 |
| **SSL** | 1 | 1 | 1 | 0 | 1 |
| **TCP** | 11 | 11 | 0 | 8 | 3 |

## VI. Conclusion

This paper presents a proposal of a system intended for troubleshooting network problems based on a passive network traffic analysis. The primary goal is to automate network diagnostics to help network administrators find causes of problems. The core of the presented approach is a multistage processing pipeline combining rule-based and inference-based methods. We have completed the implementation of a proof-of-concept system that we will use for preliminary evaluation and experiments.

We have implemented diagnostic rules for several application and service protocols. Table 1 shows the current list of supported protocols and their complexity in term of Node and Condition rule count, and their capabilities in term of Event count. After an evaluation of our solution by our partner — a monitoring vendor company, we have concluded, that the system must mark all reports which our tool may have incorrectly detected because of low-quality input data. For example, packet loss can drastically decrease the quality and accuracy of diagnostic results. In the current system, all reports from TCP flows with missing segments are marked as possibly incorrect.

Future work will focus on:

- evaluating the solution (accuracy and performance) and comparing the results with similar monitoring tools;
- analyzing each protocol's rules and based on used protocols and their field names create a filtering unit to reduce the amount of data processed by *Protocol Analyzer*;
- optimizing the performance. The current *Events Finder* combines all packets to check whether they are fulfilling the assert conditions or not. This all-to-all packet check has exponential time complexity ($2^{O(n)}$), which is unacceptable for large PCAP files. We want to optimize checking the asserts to decrease the complexity.

## Acknowledgment

This work was supported by project "Network Diagnostics from Intercepted Communication" (2017-2019), no. TH02010186, funded by the Tech-nological Agency of the Czech Republic and by BUT project "ICT Tools, Methods and Technologies for Smart Cities" (2017-2019), no. FIT-S-17-3964.

 ISBN: 978-1-61208-711-5 

## REFERENCES


[1] R. Wang, D. Wu, Y. Li, X. Yu, Z. Hui, and K. Long, "Knights tour-based fast fault localization mechanism in mesh optical communication networks," Photonic Network Communications, vol. 23, no. 2, 2012, pp. 123–129.

[2] M. Solé, V. Muntés-Mulero, A. I. Rana, and G. Estrada, "Survey on models and techniques for root-cause analysis," arXiv preprint arXiv:1701.08546, 2017.

[3] H. Zeng, P. Kazemian, G. Varghese, and N. McKeown, "A survey on network troubleshooting," Technical Report Stanford/TR12-HPNG-061012, Stanford University, Tech. Rep., 2012.

[4] M. Łgorzata Steinder and A. S. Sethi, "A survey of fault localization techniques in computer networks," Science of computer programming, vol. 53, no. 2, 2004, pp. 165–194.

[5] C. Guo et al., "Pingmesh: A large-scale system for data center network latency measurement and analysis," in ACM SIGCOMM Computer Communication Review, vol. 45, no. 4. ACM, 2015, pp. 139–152.

[6] B. Agarwal et al., "Netprints: Diagnosing home network misconfigurations using shared knowledge," in Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, ser. NSDI'09. Berkeley, CA, USA: USENIX Association, 2009, pp. 349–364.

[7] L. Lu, Z. Xu, W. Wang, and Y. Sun, "A new fault detection method for computer networks," Reliability Engineering & System Safety, vol. 114, 2013, pp. 45–51.

[8] S. Kandula et al., "Detailed diagnosis in enterprise networks," ACM SIGCOMM Computer Communication Review, vol. 39, no. 4, 2009, pp. 243–254.

[9] M. Luo, D. Zhang, G. Phua, L. Chen, and D. Wang, "An interactive rule based event management system for effective equipment troubleshooting," in IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society. IEEE, 2011, pp. 2329–2334.

[10] A. Mohamed, "Fault detection and identification in computer networks: A soft computing approach," Ph.D. dissertation, University of Waterloo, 2010.

[11] D. Brauckhoff, X. Dimitropoulos, A. Wagner, and K. Salamatian, "Anomaly extraction in backbone networks using association rules," in Proceedings of the 9th ACM SIGCOMM conference on Internet measurement. ACM, 2009, pp. 28–34.

[12] L. Benetazzo, C. Narduzzi, P. A. Pegoraro, and R. Tittoto, "Passive measurement tool for monitoring mobile packet network performances," IEEE transactions on instrumentation and measurement, vol. 55, no. 2, 2006, pp. 449–455.

[13] K.-H. Kim, H. Nam, J.-H. Park, and H. Schulzrinne, "Mot: a collaborative network troubleshooting platform for the internet of things," in Wireless Communications and Networking Conference (WCNC), 2014 IEEE. IEEE, 2014, pp. 3438–3443.

[14] T. Qiu, Z. Ge, D. Pei, J. Wang, and J. Xu, "What happened in my network: mining network events from router syslogs," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. ACM, 2010, pp. 472–484.

[15] M. Vásquez-Bermúdez, J. Hidalgo, M. del Pilar Avilés-Vera, J. Sánchez-Cercado, and C. R. Antón-Cedeño, "Analysis of a network fault detection system to support decision making," in International Conference on Technologies and Innovation. Springer, 2017, pp. 72–83.

[16] S. Jamali and M. S. Garshasbi, "Fault localization algorithm in computer networks by employing a genetic algorithm," Journal of Experimental & Theoretical Artificial Intelligence, vol. 29, no. 1, 2017, pp. 157–174.

[17] E. S. Ali and M. Darwish, "Diagnosing network faults using bayesian and case-based reasoning techniques," in Computer Engineering & Systems, 2007. ICCES'07. International Conference on. IEEE, 2007, pp. 145–150.

[18] The Wireshark Wiki, "Displayfilters," [Online; accessed 20-April-2019]. [Online]. Available: https://wiki.wireshark.org/DisplayFilters

[19] C. Xu, H. Zhang, C. Huang, and D. Peng, "Study of fault diagnosis based on probabilistic neural network for turbine generator unit," in 2010 International Conference on Artificial Intelligence and Computational Intelligence, vol. 1. IEEE, 2010, pp. 275–279.


 ISBN: 978-1-61208-711-5