# ICNS 2015

The Eleventh International Conference on Networking and Services

ISBN: 978-1-61208-404-6

**CONNET 2015**

The International Symposium on Advances in Content-oriented Networks and Systems

May 24 - 29, 2015

Rome, Italy

**ICNS 2015 Editors**

Eugen Borcoci, University 'Politehnica' Bucharest, Romania

Tao Zheng, Orange Labs Beijing, China

Carlos Becker Westphall, University of Santa Catarina, Brazil

# ICNS 2015

# Forward

The Eleventh International Conference on Networking and Services (ICNS 2015), held between May 24-29, 2015 in Rome, Italy, 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 also considered.

IPv6, the Next Generation of the Internet Protocol, has seen over the past 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 refer to high performance and computability in Grid networks. However, the communication and computation mechanisms for Grid applications require further development, deployment and validation.

The conference had the following tracks:

- Multi-technology service deployment and assurance
- CLOUD/GRID Networks and Services
- Emerging Network Communications and Technologies
- Next Generation Networks and Software Defined Networking

The conference also featured the following symposium:

- **CONNET 2015, *The International Symposium on Advances in Content-oriented Networks and Systems***

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the ICNS 2015 technical program committee, as well as the numerous reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to ICNS 2015. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ICNS 2015 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope ICNS 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of networking and services. We also hope that Rome, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the historic beauty of the city.

**ICNS 2015 Chairs**

**ICNS Advisory Chairs**

Pedro Andrés Aranda Gutiérrez, Telefónica I+D - Madrid, Spain
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Eugen Borcoci, University 'Politehnica' Bucharest, Romania
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Simone Silvestri, Missouri University of Science and Technology, USA
Yoshiaki Taniguchi, Kindai University, Japan
Go Hasegawa, Osaka University, Japan
Abdulrahman Yarali, Murray State University, USA
Emmanuel Bertin, Orange Labs, France
Steffen Fries, Siemens, Germany
Rui L.A. Aguiar, University of Aveiro, Portugal
Iain Murray, Curtin University of Technology, Australia
Khondkar Islam, George Mason University - Fairfax, USA

**ICNS Industry/Research Relation Chairs**

Eunsoo Shim, Samsung Electronics, Korea
Tao Zheng, Orange Labs Beijing, China
Bruno Chatras, Orange Labs, France
Jun Kyun Choi, KAIST, Korea
Michael Galetzka, Fraunhofer Institute for Integrated Circuits - Dresden, Germany
Mikael Gidlund, ABB, Sweden
Juraj Giertl, T-Systems, Slovakia
Sinan Hanay, NICT, Japan

**CONNET 2015 Advisory Committee**

Eugen Borcoci, University Politehnica of Bucharest, Romania
Pascal Lorenz, University of Haute Alsace, France
Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany

# ICNS 2015

# Committee

## ICNS 2015 Advisory Chairs

Pedro Andrés Aranda Gutiérrez, Telefónica I+D - Madrid, Spain
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Eugen Borcoci, University 'Politehnica' Bucharest, Romania
Jaime Lloret Mauri, Polytechnic University of Valencia, Spain
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Simone Silvestri, Missouri University of Science and Technology, USA
Yoshiaki Taniguchi, Kindai University, Japan
Go Hasegawa, Osaka University, Japan
Abdulrahman Yarali, Murray State University, USA
Emmanuel Bertin, Orange Labs, France
Steffen Fries, Siemens, Germany
Rui L.A. Aguiar, University of Aveiro, Portugal
Iain Murray, Curtin University of Technology, Australia
Khondkar Islam, George Mason University - Fairfax, USA

## ICNS 2015 Industry/Research Relation Chairs

Eunsoo Shim, Samsung Electronics, Korea
Tao Zheng, Orange Labs Beijing, China
Bruno Chatras, Orange Labs, France
Jun Kyun Choi, KAIST, Korea
Michael Galetzka, Fraunhofer Institute for Integrated Circuits - Dresden, Germany
Mikael Gidlund, ABB, Sweden
Juraj Giertl, T-Systems, Slovakia
Sinan Hanay, NICT, Japan

## ICNS 2015 Technical Program Committee

Johan Åkerberg, ABB AB - Corporate Research - Västerås, Sweden
Ryma Abassi, Higher School of Communication of Tunis /Sup'Com, Tunisia
Nalin Abeysekera, University of Colombo, Sri Lanka
Ferran Adelantado i Freixer, Universitat Oberta de Catalunya, Spain
Hossam Afifi, Télécom SudParis | Institut Mines Télécom, France
Prathima Agrawal, Auburn University, USA
Javier M. Aguiar Pérez, Universidad de Valladolid, Spain

Rui L.A. Aguiar, University of Aveiro, Portugal
Mehmet Akşit, University of Twente, Netherlands
Basheer Al-Duwairi, Jordan University of Science and Technology, Jordan
Ali H. Al-Bayatti, De Montfort University - Leicester, UK
Maria Andrade, University of Porto / INESC Porto, Portugal
Annamalai Annamalai, Prairie View A&M University, USA
Mario Anzures-García, Benemérita Universidad Autónoma de Puebla, Mexico
Pedro Andrés Aranda Gutiérrez, Telefónica I+D - Madrid, Spain
Patrick Appiah-Kubi, Indiana State University, USA
Bourdena Athina, University of the Aegean, Greece
Isabelle Augé-Blum, CITI, INSA-Lyon / Urbanet, INRIA, France
Mohamad Badra, Zayed University, United Arab Emirates
Aleksandr Bakharev, Siberian State University of Telecommunication and Information Sciences, Russia
Mohammad M. Banat, Jordan University of Science and Technology, Jordan
Javier Barria, Imperial College of London, UK
Mostafa Bassiouni, University of Central Florida, USA
Michael Bauer, The University of Western Ontario - London, Canada
Carlos Becker Westphall, Federal University of Santa Catarina, Brazil
Tarek Bejaoui, University of Carthage, Tunisia
Mehdi Bennis, University of Oulu, Finland
Luis Bernardo, Universidade Nova de Lisboa, Portugal
Emmanuel Bertin, Orange Labs, France
Eugen Borcoci, University "Politehnica"of Bucharest (UPB), Romania
Fernando Boronat Seguí, Polytechnic University of Valencia, Spain
Pierre Boulanger, University of Alberta, Canada
Kalinka Branco, University of São Paulo, Brazil
Dumitru Burdescu, University of Craiova, Romania
Jens Buysse, Ghent University/IBBT, Belgium
Maria Dolores Cano Baños, Polytechnic University of Cartagena - Campus Muralla del Mar, Spain
Juan Carlos Cano, DISCA - Universitat Politècnica de València, Spain
Tarik Caršimamovic, BHTelecom, Bosnia and Herzegovina
José Cecílio, University of Coimbra, Portugal
Patryk Chamuczyński, Radytek, Poland
Bruno Chatras, Orange Labs, France
Jun Kyun Choi, KAIST, Korea
Victor Clincy, Kennesaw State University, USA
Jorge A. Cobb, University of Texas at Dallas, USA
Hugo Coll Ferri, Universidad Politecnica de Valencia, Spain
Todor Cooklev, Indiana University - Purdue University Fort Wayne, USA
Alejandro Cordero, Amaranto Consultores, Spain
Taiping Cui, Inha University - Incheon, Korea
João Henrique de Souza Pereira, University of São Paulo, Brazil

Zhihong Hong, Communications Research Centre, Canada
Per Hurtig, Karlstad University, Sweden
Naohiro Ishii, Aichi Institute of Technology, Japan
Khondkar Islam, George Mason University - Fairfax, USA
Arunita Jaekel, University of Windsor, Canada
Tauseef Jamal, SITILab Lisbon, Portugal
Peter Janacik, University of Paderborn, Germany
Imad Jawhar, United Arab Emirates University, UAE
Ravi Jhawar, Universitàdegli Studi di Milano - Crema, Italy
Sudharman K. Jayaweera, University of New Mexico - Albuquerque, USA
Ying Jian, Google Inc, USA
Fan Jiang, Tuskegee University, USA
Wei Jiang, Missouri University of Science and Technology, USA
Eunjin (EJ) Jung, University of San Francisco, USA
Maxim Kalinin, St. Petersburg State Polytechnical University, Russia
Enio Kaljic, University of Sarajevo, Bosnia and Herzegovina
Georgios Kambourakis, University of the Aegean - Karlovassi, Greece
Hisao Kameda, University of Tsukuba, Japan
Kyungtae Kang, Hanyang University, Korea
Nirav Kapadia, Public Company Accounting Oversight Board (PCAOB), USA
Georgios Karagiannis, University of Twente, The Netherlands
Masoumeh Karimi, Technological University of America, USA
Hiroyuki Kasai, University of Electro-Communications, Japan
Aggelos K. Katsaggelos, Northwestern University - Evanston, USA
Sokratis K. Katsikas, University of Piraeus, Greece
Thomas Kemmerich, University College Gjøvik, Norway
Razib Hayat Khan, NTNU, Norway
Ki Hong Kim, The Attached Institute of ETRI, Korea
Younghan Kim, Soongsil University - Seoul, Republic of Korea
Mario Kolberg, University of Stirling - Scotland, UK
Lisimachos Kondi, University of Ioannina, Greece
Jerzy Konorski, Gdansk University of Technology, Poland
Elisavet Konstantinou, University of the Aegean, Greece
Kimon Kontovasilis, NCSR "Demokritos", Greece
Andrej Kos, University of Ljubljana, Slovenia
Evangelos Kranakis, Carleton University, - Ottawa, Canada
Diego Kreutz, University of Lisbon, Portugal
Francine Krief, University of Bordeaux, France
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Anju Lata Yadav, Shri G. S. Institute of Technology and Science, India
Suk Kyu Lee, Korea University at Seoul, Republic of Korea
DongJin Lee, Auckland University, New Zealand
Leo Lehmann, OFCOM, Switzerland
Ricardo Lent, Imperial College London, UK

Stephan Trahasch, Hochschule Offenburg, Germany
Joseph G. Tront, Virginia Tech, USA
Binod Vaidya, University of Ottawa, Canada
Geoffroy R. Vallee, Oak Ridge National Laboratory (ORNL), USA
Fabrice Valois, INSA Lyon, France
Hans van den Berg, TNO / University of Twente, The Netherlands
Ioannis O. Vardiambasis, Technological Educational Institute (TEI) of Crete - Branch of Chania, Greece
Vladimir Vesely, Brno University of Technology, Czech Republic
Dario Vieira, EFREI, France
Bjørn Villa, Norwegian Institute of Science and Technology, Norway
José Miguel Villalón Millan, Universidad de Castilla - La Mancha, Spain
Demosthenes Vouyioukas, University of the Aegean - Karlovassi, Greece
Arno Wacker, University of Kassel, Germany
Bin Wang, Wright State University - Dayton, USA
Junwei Wang, University of Tokyo, Japan
Mea Wang, University of Calgary, Canada
Tingkai Wang, London Metropolitan University, UK
Michelle Wetterwald, HeNetBot, France
Alexander Wijesinha, Towson University, USA
Ouri Wolfson, University of Illinois - Chicago, USA
Zhengping Wu, University of Bridgeport, USA
Feng Xia, Dalian University of Technology, China
Serhan Yarkan, Istanbul Commerce University, Turkey
Homayoun Yousefi'zadeh, University of California - Irvine, USA
Vladimir S. Zaborovsky, Polytechnic University/Robotics Institute - St.Petersburg, Russia
Sherali Zeadally, University of the District of Columbia, USA
Jie Zeng, Tsinghua University, China
Tao Zheng, Orange Labs Beijing, China
Yifeng Zhou, Communications Research Centre, Canada
Ye Zhu, Cleveland State University, USA
Yingwu Zhu, Seattle University, USA
Piotr Zuraniewski, University of Amsterdam (NL), The Netherlands /AGH University of Science and Technology, Poland

**CONNET 2015 Advisory Committee**

Eugen Borcoci, University Politehnica of Bucharest, Romania
Pascal Lorenz, University of Haute Alsace, France
Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany

## CONNET 2015 Technical Program Committee

Mustafa Ilhan Akbas, University of Central Florida, USA
Bogdan Ghita, Plymouth University, UK
Krishna Kant, Temple University, USA
Shen Li, University of Illinois at Urbana-Champaign, USA
Hengchang Liu, University of Science and Technology of China, China
Pascal Lorenz, University of Haute Alsace, France
Andreas Merentitis, National and Kapodistrian University of Athens, Greece
Marek Miskowicz, AGH University of Science and Technology, Poland
Gregory O’Hare, University College Dublin, Ireland
Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster / Leibniz Universität Hannover / North-German Supercomputing Alliance, Germany
Javid Taheri, University of Sydney, Australia
Hideki Tode, Osaka Prefecture University, Japan

**Copyright Information**

## Table of Contents

# An Automated Framework for Command and Control Server Connection and Malicious Mail Detection

Lo-Yao Yeh
Department of Network and Security
National Center for High-Performance Computing
Taichung, Taiwan (ROC)
e-mail:lyyeh@narlabs.org.tw

Yi-Lang Tsai
Department of Director
National Center for High-Performance Computing
Tainan, Taiwan (ROC)
e-mail:yilang@narlabs.org.tw

***Abstract*—In recent Internet development, the amount of malware has increased significantly. There are more and more methods that hackers can use to infect personal computers to send spam mails, steal personal information, and launch Distributed Denial of Service (DDoS) attacks. This paper proposes a framework to strengthen security for users by integrating several online resources. The proposed framework can automatically prevent users from visiting malicious websites on the Internet Explorer browser. In addition, it can automatically detect the mail's source and attached files. Finally, if malware is connected to any Command and Control (C&C) servers, our framework is able to detect it by using an Application Programming Interface (API) hooking technique, and automatically kill it. By these methods, it will effectively restrain the scale of botnets and significantly reduce the risk of personal computers infection.**

***Keywords—Network Security; Botnet; Email; API hooking;***

## I. INTRODUCTION

As long as the number of computer devices increases, so does the number of compromised computers. If a device does not have any defense mechanisms, it will easy to become a bot. There are many ways for hackers to spread malware. For example, they can set up a malicious web site and use various means to induce individuals to browse it. Hackers can also attack normally benign websites. Most people worry less about benign websites, so they will generally trust all the files on the sites and perhaps download them. Alternatively, email is another popular way for hackers to launch attacks. Hackers can use email to spread links to users, luring them to phishing web sites, or masquerade as well-known companies to deliver emails to trick users and steal user account and password information. In addition, email attachments may also contain malicious files. When a user inadvertently executes the malware, their personal computer is turned into a bot and becomes part of a botnet. The bot behavior includes stealing personal information, sending spam and viruses, or launching Denial of Service (DoS) attacks. Therefore, the rapid expansion of botnets must be limited, and the user must have a security system to reduce their risks. This paper integrates the National Center for High-Performance Computing (NCHC) blacklist database [1], Virustotal [2], and the WHOIS server to reduce the probability of a user's computer becoming a bot. By parsing packets, our framework can automatically block a suspect page and warn users when they try to browse a known malicious webpage or download a malicious file. The analysis of web mail headers is used to determine whether or not mail is malicious, therefore protecting users. Finally, our system can monitor all applications on the PC to detect if any applications are trying to connect to a known Command and Control (C&C) server, which is used by a hacker to control the bots. Hence, we can effectively discover the potential malware and minimize the risk of infection.

## II. PROPOSED SYSTEM ARCHITECTURE

Today, Microsoft Windows is the most popular operation system in the world, so our system architecture is designed for implementation on Windows XP and Windows 7. Our framework, as shown in Figure 1, is divided into two components, namely, the daemon program and the toolbar program. The daemon program monitors the network device to collect packets and uses the hooking technique to examine applications on user's computer. The toolbar is responsible to get the page information from the Internet Explorer (IE) browser.

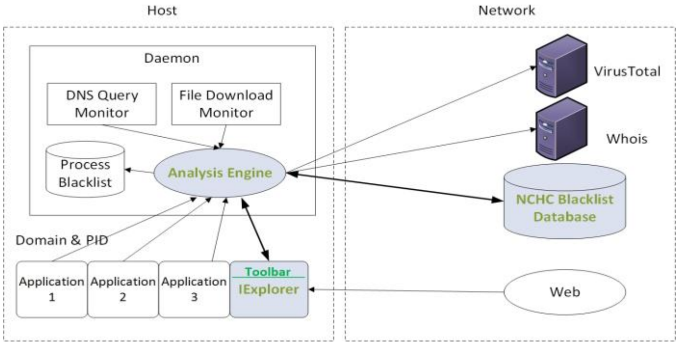


Figure 1. System Architecture

### A. *Analysis Engine*

Our system integrates three different resources to complement our framework, including the online resource VirusTotal, the NCHC blacklist database, and the WHOIS server. Our framework uses the NCHC blacklist database and VirusTotal to check whether or not the domain or the files are malicious. In addition, our framework also takes advantage of WHOIS servers to detect forged mails.

### B. *Monitor Processes*

Usually, malware will resolve Domain Name System (DNS names to locate the C&C server [4]. In our previous version of such a framework [5], our system parsed packets to check collected domains with the NCHC blacklist database. If a process connected to a known malicious domain, our system notified the user immediately. However, the drawback of our previous system is that we were unable to identify which process launched the connection. To solve this problem, we use

EasyHook [3] to integrate our system and hook each application. The daemon program can obtain domain names and Process IDentifier (PID) when the hooked process connects to any domain.

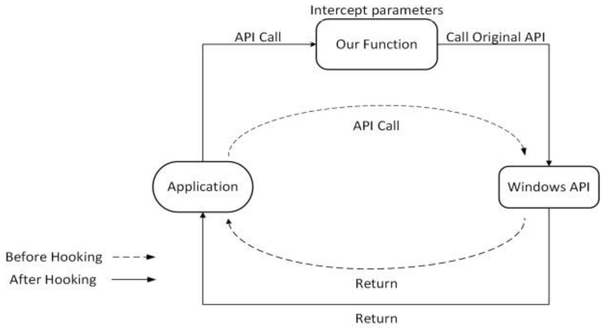


Figure 2. Execution flow with/without hooking.

When a process wants to resolve domain name service (DNS) names, it calls the specific API function to achieve this. Our system uses Easyhook to inject the pre-established data link layer or Dynamic Link Library (DLL) into every application's memory space. The DLL includes our function that intercepts parameters of the specific API function call, one of which is domain name. Figure 2 shows the execution flow of the function call before and after API hooking. The solid lines represent an execution flow without hooking. The dotted lines represent an execution flow with hooking. After the intercept parameters, we must to call original API to allow the application to finish its work. If any process connects to a C&C server, our system can obtain the malicious domain and PID from the process to show which specific process tried to connect to the malicious domain. The user can determine which one may be the malware and decide to kill this or not.

### C. *Web detection*

- Blocking Malicious Pages

Our system can notify users when they are visiting malicious pages. Furthermore, the system can also actively block the malicious page before users visit it. The daemon commands every toolbar to monitor its own page. When the page wants to connect to a malicious web site, the toolbar blocks the page and asks users if they really want to visit the known malicious web site. If the answer is negative, the toolbar redirects the page to a blank page.

- Web Mail Detection

Email is one of the popular methods for hackers to attack computers. To protect users, our system has the following functions to detect malicious emails, and can be implemented on a web mail service.

#### *1) Email Authentication*

For mail reliability, most well-known companies, like Gmail and Yahoo! mail, use SPF [6] and DKIM [7] for mail authentication and spam filters. When the mail server receives mail, it will validate the identity of the sender, and then add an *authentication-results* header to the mail header. Our system can then check the mail header to examine results of SPF and DKIM authentication. In other words, when users receive mails from a well-known company, but the mail does not pass this authentication, users can be warned that the mail may be forged.

#### *2) Mail Attachment*

In order to prevent users from downloading malware from email, our system also examines the attachments. If a mail has attachments, it represents the sender requesting the content to be saved as a file. We can obtain the original files by decoding the mail body. Finally, our system will automatically upload the attached files to VirusTotal and the NCHC server to check whether the received files are malicious.

#### *3) Received header*

The *received* header is the most important header for tracking mail. When the mail server receives the message, the server adds a *received* header to the top of the header, recording the sender's name and IP address. However, because the sender's name can be manipulated, our system reads the *received* headers from top to bottom to detect whether the mail is forged. First, we check the sender's IP address item in each *received* header by WHOIS server to ensure that its domain name and sender's name are the same. Second, our system confirms whether they are malicious domains in the NCHC blacklist database. However, because some mail servers do not use their domain as the server name, we provide two further pieces of information, the *Alexa Rank* and *Page Views per Visit*, as a simple way for users to check the validity of the domain. In general, malicious domains unlikely have a high Alexa ranking and low page view value.

## V. CONCLUSION

When using our framework, if a user connects to a known malicious domain, an infection alarm is issued to warn the user about the potential threat. Moreover, if malware already installed in the computer connects to the domain of a C&C server, our system can also find and kill the process. For email protection, our system not only detects the mail's source, but automatically scans attached files. As a result, we can effectively restrain the influence of botnets and reduce the chances of a PC becoming infected.

## REFERENCE

[1] NCHC malware knowledge base. [Online]. Available from: http://owl.nchc.org.tw, 2015.03.25.

[2] VirusTotal. [Online]. Available from: https://www.virustotal.com/en/, 2015.03.25.

[3] EasyHook. [Online].Available from:http://easyhook.codeplex.com/, 2015.03.25.

[4] S. Shin, Z. Xu, and G. Gu, "Effort: Efficient and effective bot malware detection," Proc. IEEE INFOCOM, 2012, pp. 2846–2850.

[5] L.-Y. Yeh, Y.-L. Tsai, B.-Y. Lee, and J.-G. Chang "An Automatic Botnet Detection and Notification System in Taiwan", Intnational Conf. Security and Management, 2013, pp. 469-471.

[6] RFC 6652, Sender Policy Framework (SPF) Authentication Failure Reporting Using the Abuse Reporting Format, Proposed Standard, 2012.

[7] RFC 6376, DomainKeys Identified Mail (DKIM) Signatures Draft Standard, 2011.

# DropTail Based ConEx Applied to Video Streaming

Ali Sanhaji, Philippe Niger, Philippe Cadro
Orange Labs
2, avenue Pierre Marzin,
22300 Lannion, France
Email: {ali.sanhaji, philippe.niger, philippe.cadro}@orange.com

André-Luc Beylot
Toulouse Institute of Computer Science Research (IRIT)
INP-ENSEEIHT, Laboratoire IRIT,
2, rue Charles Camichel,
31071 Toulouse Cedex 7, France
Email: andre-luc.beylot@enseeiht.fr

***Abstract*—With Internet traffic ever increasing, network congestion should occur more and more frequently. During congestion periods, some users contribute more than others to the congestion in the network. It might be interesting for a network operator to differentiate between users proportionally to the congestion they induce, but the necessary information for this purpose is not available at the network layer, and is exchanged at the transport layer (e.g., Transmission Control Protocol (TCP) acks). This led the Internet Engineering Task Force (IETF) to design Congestion Exposure (ConEx), a new mechanism to expose to the network the amount of congestion a user is responsible for. However, ConEx needs other mechanisms such as Random Early Detection (RED), Explicit Congestion Notification (ECN) and a number of modifications to the senders and receivers to be fully operational. Nonetheless, it is deployable with few modifications by relying only on loss information in DropTail queues to improve the fairness between users. The aim of this paper is to provide a comparison between the performance in terms of fairness improvement provided by ConEx with few modifications and by ConEx with complete modifications. Firstly, we will see that despite the limited accuracy due to the few changes, ConEx still provides good fairness improvement between users. Secondly, we will discuss the weaknesses ConEx presents with regard to short-lived flows. Finally, we will show how ConEx can help during congestion periods to enhance the Quality of Experience (QoE) of video streaming users (based on a YouTube traffic model).**

***Keywords*-ConEx; ECN; Congestion; policing; YouTube; LEDBAT.**

## I. INTRODUCTION

During the network's busy hours, a greater amount of traffic than what the network can handle leads to congestion, affecting the quality of experience of many users. Yet, this great amount of traffic is mainly caused by a small percentage of users. For example, in Orange's Fiber To The Home (FTTH) access networks, 80% of downstream traffic is generated by 15% of the customers [1]. The aim is to convince these heavy users to yield network resources during congestion periods for the benefit of everybody.

Some traffic management approaches are already implemented like rate-limiting or defining Data-Volume caps above which the users are slowed down or stopped. However, these solutions show limited efficiency because they do not consider the network state, if it is congested or not, if the rate-limited user has seriously hampered the experience of the others, if this "heavy user" yielded the network resources when encountering congestion. A heavy user might consume his allowed Data-Volume even when the network is not in a congestion phase. It would be fairer to limit the users according to how much congestion they induced. For this, we would need the information about the congestion encountered by the users. This valuable congestion information can be exchanged between the users at the transport layer (e.g., through TCP acks) but it is transparent for the network layer.

To counter this lack of information at the network layer, the IETF designed ConEx, which is a mechanism that allows the sender to inform the network about the congestion encountered [2]. The amount of lost and ECN-marked packets exposed by a user defines a new metric called the **Congestion-Volume**, which is a more useful metric than **Data-Volume** because it reports directly the congestion in the network.

The implementation of ConEx relies on existing mechanisms like RED, ECN capability on routers and new features to both the sender and the receiver to be fully ConEx-capable. We are interested in whether or not ConEx still presents a good performance without the use of ECN and relying only on minimal modifications to the users. In this paper, we will first present in Section II the related work on ConEx. Section III will describe the ConEx principle and the mechanisms on which it relies. The performance evaluation of ConEx with and without ECN using long-lived flows is presented in Section IV while the short-lived flows issue will be discussed in Section V. Our interest will be focused, in Section VI, on how ConEx can be useful in the case of video streaming traffic to enhance the users' QoE, with scenarios using a YouTube traffic model, and how heavy users can take advantage in using a congestion control algorithm like Low Extra Delay Background Transport (LEDBAT). Section VII summarises the contributions, finally, Section VIII discusses the future work, still waiting to be covered.

## II. RELATED WORK

The IETF launched a working group to develop experimental specifications of ConEx in IPv6 networks [2]. An Request For Comments (RFC) [3] discussing the concepts and use cases has been published, and other drafts concerning the ConEx mechanism are currently available: the use of a destination option in the IPv6 Header to carry the ConEx markings and the necessary modifications to TCP [4].

Re-ECN is a "pre-ConEx" implementation solution to allow congestion exposure for IPv4 networks. A thorough description and analysis of the Re-ECN mechanism has been done under the Trilogy project [5]. This work had a great influence for the emergence of the ConEx working group.

Some papers focused on the performance evaluation of the congestion exposure mechanism through the evaluation of Re-ECN in multiple scenarios. [6] developed a Linux implementation of Re-ECN and performed several simulations showing the great dependency of the Re-ECN information to the flow size, the Round Trip Time (RTT) and the Active Queue Management (AQM) parameters. [7] evaluates mobility issues with congestion exposure and shows that mobility is not a major concern for Re-ECN. [8] evaluates Re-ECN applicability in LTE networks and found that it can bring a significant improvement for these networks unless they are under severe packet loss rate. All these papers rely on the use of ECN to signal congestion; to our knowledge, no performance evaluation of ConEx has been made solely based on loss exposure.

## III. Congestion Exposure

In this section, we will describe ConEx, how it operates to expose congestion, along with the other mechanisms used to collect congestion information and control the users' traffic.

### A. *ConEx mechanism*

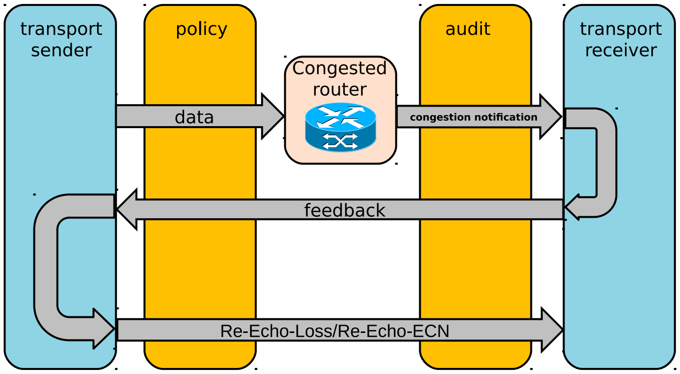


Figure 1. ConEx mechanism

Figure 1 shows the whole ConEx process and all the elements involved with it. The ConEx mechanism works as follows: A transport sender starts by sending a data packet in the network, this packet might encounter one or several congested routers along its path. The packet will either be lost or ECN marked (by setting the Congestion Experienced (CE) codepoint in the IP header [10]) by the congested routers. This information about loss or marking will reach the transport receiver, and through the TCP acknowledgments, the receiver will feedback this information to the sender. With the use of ConEx, the sender will reinject this feedback to the network in the IP packet headers (e.g., use of the RE bit in section III-D), which will hold the Re-Echo signals. Detecting a loss will generate a Re-Echo-Loss signal from the sender, while an ECN marked packet will generate a Re-Echo-ECN signal.

The information provided by ConEx can then be used by the network operator for traffic management through a congestion policer for example. At the ingress of the network, a congestion policer counts the congested packets and takes traffic control policy decisions (e.g., discard, deprioritize packets using Differentiated Services (DiffServ)) if the user has consumed the congestion-volume he was allowed. At the egress of the network, an auditor makes sure that the senders are exposing the right amount of congestion in the network. It helps as a prevention from the users understating the congestion their flows encounter, but if the sources are trusted ones, the auditor is unnecessary.

### B. *Random Early Detection*

Random Early Detection is an Active Queue Management technique, implemented on many routers, which was first introduced in [9]. It allows to randomly drop or ECN mark packets according to a probability which increases from 0 to the maximum probability $p_{max}$ when the mean queue length increases from a minimum threshold to a maximum threshold. Above the maximum threshold, all packets are either dropped or marked if ECN is used.

### C. *Explicit Congestion Notification*

Explicit Congestion Notification [10] is a way to indicate the occurrence of congestion in the network without having to drop packets. It uses two bits [ECT,CE] of the IP header to signal congestion to the receiver.

### D. *Re-ECN*

Re-ECN is a candidate implementation of ConEx for IPv4 [5]. It uses the bit 48 (RE bit) of the IPv4 header to extend the ECN field to a 3-bit field, allowing 8 codepoints. These codepoints identify the ConEx signals as described in Table I.

TABLE I. ConEx signals with Re-ECN encoding

| ECN field | RE bit | ConEx signal |
|---|---|---|
| 00 | 1 | Credit (Used with the auditor) |
| 01 | 1 | ConEx-Not-Marked (ConEx-Capable) |
| 01 | 0 | Re-Echo-ECN or Re-Echo-Loss |
| 11 | 1 | ECN marked packet |
| 11 | 0 | Re-Echo packet and ECN-marked |
| 10 | 0 | ECN legacy (Not-ConEx) |
| 00 | 0 | Not-ECN (Not-ConEx) |
| 10 | 1 | Unused |

### E. *TCP modifications*

The classic ECN mechanism as described in [10] allows the receiver to feedback only one CE mark per RTT. Indeed, even if several packets of the same flow get CE marked during one RTT, the receiver has only one bit (ECN-Echo (ECE) flag in the TCP header) to feedback all the marks. The information about how many packets have been marked is valuable for ConEx but also for other mechanisms like DCTCP [11]; modifications to TCP are needed to provide more than one feedback per RTT. [12] proposes a solution to achieve such a goal. It suggests to overload the three TCP flags ECE, Congestion Window Reduced (CWR) and Nonce Sum (NS) to form a 3-bit field. This field would act as a counter for the number of CE marks seen by the receiver which can feedback it to the sender, allowing the sender to follow the accurate evolution of ECN markings and report the right amount of Re-Echo-ECN signals.

### F. *Congestion policer*

The great advantage with ConEx is to allow the network operator to police the users proportionally to their contribution to congestion, thus to the impact they have on other users. The policing can be applied at the ingress to prevent the heavy users from overloading the network. The congestion policer can be implemented as a token bucket with a filling rate $r$ (the allowed Congestion-Rate) and a depth $d$ (the allowed Congestion-Burst). The policer removes the same amount of tokens from the bucket as there are bytes in the Re-Echo-ECN/Re-Echo-Loss packets sent by a user. When the bucket empties, the policer proceeds to discard the packets of the user who exceeded his allowed Congestion-Volume. As shown in Figure 2, there are three levels of policing used in the

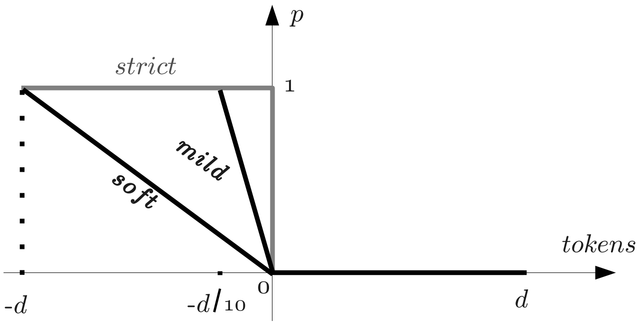


Figure 2. Drop function of the congestion policer

performance evaluation, the strict policer discards all packets when the bucket is empty, the mild policer discards packets with a probability increasing from 0 to 1 when the bucket depth decreases from 0 to $-d/10$ and the soft policer discards packets with a probability increasing from 0 to 1 when the bucket depth decreases from 0 to $-d$.

## IV. Long-lived flows

### A. Simulated Network

To perform the simulations, we used the Network Simulator 2 (NS2) [13] in which we implemented ConEx following the latest RFCs and drafts and we used the IPv4 proposal presented in Section III-D. The simulated network is depicted in Figure 3. There are 100 users on either side of the network, each single user on the right receiving traffic from a single user on the left. 90 of them are light users using only one File Transfer Protocol (FTP) flow each as a traffic source. The other 10 users are heavy users, they use 36 FTP flows each as a traffic source, they will thereby be responsible for 80% of the traffic on the bottleneck. The TCP senders use cubic as a congestion control algorithm with Selective Acknowledgments (SACK) and TimeStamps options. The TCP receivers can feedback ECN markings in a accurate count to the sender which in turn will send a Re-Echo-ECN/Re-Echo-Loss signal for every ECN-marked/lost packet. The TCP maximum window value is equal to 64KB while the packet size is equal to 1500 bytes.

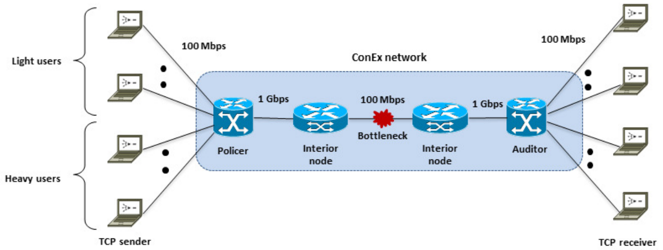


Figure 3. Simulated network topology

All users have 100ms Round Trip Time and share a 100Mbps bottleneck. At the ingress of the network, there is a per user congestion policer, which is implemented as described in Section III-F. The action taken by the policer is dropping the user's packets when the bucket, which has a depth of 64KB, is emptied. On the bottleneck's router, there is a RED queue with a length equal to the Bandwidth Delay Product (BDP) in order to hold 100ms of the bottleneck's packets. The probability of marking packets increases from 0 to $p_{max} = 1$ as the average queue length increases from 10% to 100% of the total queue length. At the egress of the network, there is an auditor which is deactivated because we use trusted sources.

A single simulation lasts 100s and is run 30 times to have proper 95% confidence intervals for each point. For greater visibility of the graphs, these intervals are not depicted when their value is around 1% of the metric's mean. The traffic sources are saturated and each flow starts randomly between 0 and 300ms.

TCP provides a flow-based fairness, meaning that a user can get more bandwidth share if he uses more flows. The per user congestion policer does not consider the user's flows individually but only the aggregate traffic of the user to monitor the amount of congestion induced in the network. The purpose of ConEx is to improve fairness between users, especially between the light user and the heavy user. Therefore, we will be monitoring a metric defined in [14]:

$$unfairness = \frac{throughput\ of\ a\ heavy\ user}{throughput\ of\ a\ light\ user} \tag{1}$$

Through the action of the policer, ConEx provides the ability to decrease the unfairness between users in a congested network. In the following sections, we will discuss the impact of the filling rate and the harshness of the congestion policing. Afterwards, we will compare the performance of ConEx when all the modifications are applied with the performance of ConEx when only the minimum modifications are applied.

### B. Policer harshness

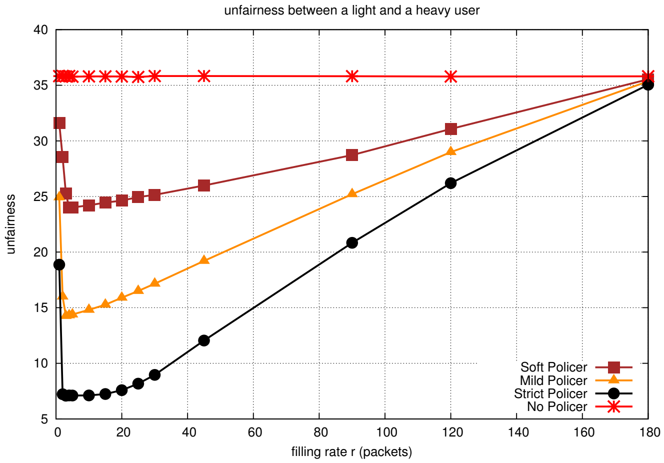


Figure 4. Unfairness between a heavy and a light user

Figure 4 represents the average unfairness versus the allowed filling rate of a user in the simulation. Each curve represents a level of harshness of the policer as explained in Section III-F. The straight red curve on top is the unfairness when no policing is applied (the policer is deactivated). Only TCP is performing congestion control and TCP induces fairness between flows; as a heavy user has 36 flows and a light user has only one, the unfairness is equal to 36 as expected. When the policer is activated (the three remaining curves), the heavy users are the ones that will be the most policed. As the heavy users are forced to reduce their throughput, the light users occupy the freed bandwidth and the unfairness is reduced.

In Figure 4, the unfairness presents a minimum value suggesting an optimal filling rate. On the two sides of the optimum, the unfairness increases but for two different reasons. On the right side, as the filling rate increases, the heavy users undergo less policing. They get a higher throughput than with the optimal filling rate and the unfairness increases. When the filling rate is high enough, the heavy users avoid the policer's intervention, so the unfairness reaches the value obtained without policing ($unfairness = 36$). On the left side of the optimum, both the heavy users and the light users are policed because of the insufficient filling rate. The light users are forced to reduce their throughput and the unfairness increases compared to the unfairness with the optimal rate. Policing the light users is counter-productive if the purpose is to reduce unfairness between light and heavy users; one has to attribute filling rates which will avoid the light users from being policed while keeping the heavy users from overloading the network during busy hours.

To evaluate the impact of the harshness of the policer, a soft, a mild and a strict policer are used, which drop packets with increasing aggressiveness. Figure 4 shows that the three policers present the same optimal filling rate but are different in decreasing the unfairness. The harsher is the policing, the lower is the unfairness, because the heavy users will need to further reduce their throughput due to the policer's higher dropping probability. The difference between the policers is substantial because when the policer drops packets, ConEx will react by sending more Re-Echo-Loss packets which will eventually lead to more policing. With a severe policer, the risk is to have a user continually decreasing his throughput because of the policer's actions while the network is uncongested. This fact should be taken into account in the design of the policer's algorithm.

### C. ConEx with increasing complexity

The deployability of ConEx is a major concern for a network operator, and ConEx allows incremental deployement by requiring only a few modifications to be operational. It can afterwards be upgraded to provide a more accurate feedback of congestion information.

TABLE II. ConEx with increasing complexity

| Case | queue | sender | receiver |
|---|---|---|---|
| DTConEx | DropTail | No ECN | No ECN |
| REDConEx | RED | No ECN | No ECN |
| FullConEx | RED | Accurate ECN | Accurate ECN |

The minimum modifications needed for ConEx are the modifications of the sender which will react to a loss detection by sending a Re-Echo-Loss signal. In this case, ECN support is needed neither on the sender nor on the receiver and the RED queue can be replaced by a simple DropTail queue, which will drop packets when it overflows. In the next paragraphs, this case is referred as the $DTConEx$ case. The next step of modifications is when a RED queue is used on the router to improve reactivity to congestion appearance. ECN is not used and ConEx will react only to dropped packets by the RED queue. This is referred as the $REDConEx$ case. The ultimate step of modifications is when ECN is used by both the sender and the receiver along with modifications to the receiver to allow accurate ECN feedback. This is referred as the $FullConEx$ case. The three cases are summurised in Table II.

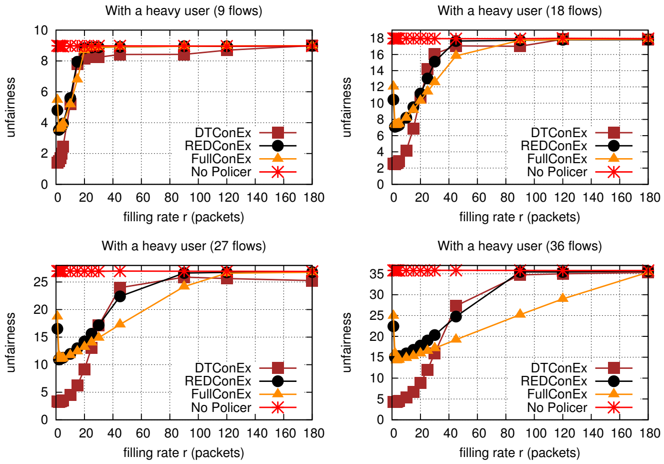


Figure 5. Unfairness with DTConEx, REDConEx and FullConEx

Figure 5 depicts the average unfairness versus the filling rate in four scenarios where we vary the number of flows per heavy user (9, 18, 27, 36), while the light user remains with a single flow. In each scenario, the red curve represents the unfairness without policing, while the three other curves represent the three cases explained above. As the number of flows of a heavy user increases, the number of Re-Echo packets sent increases, consuming more tokens, leading to more policing, so the range of filling rates allowing fairness improvement is widened.

In all scenarios, we see that $FullConEx$ decreases the unfairness more than $REDConEx$. The reason is that the former case provides both the information on ECN and on losses, which makes the policer more accurate in its actions.

The $DTConEx$ case provides even less congestion information than the two other cases but manages to decrease more the unfairness in all scenarios in a range of filling rates around the optimum. $DTConEx$ is effective because it does not make the light users reduce their throughput as early as the two other cases. Indeed, in both $REDConEx$ and $FullConEx$, the queue drops or marks packets when its mean length exceeds a minimum threshold forcing the users to reduce their throughput. The DropTail queue only drops packets when the entire queue is filled, which gives the opportunity for the light users to increase their throughput when heavy users are restrained by the policer.

Figure 6 represents the mean queueing delay and the loss rate that a light user encounters as a function of the filling rate (scenario with 36 flows per heavy user). A DropTail queue does not allow, unlike RED, to reduce the queueing delay observed by the users as we can see in the $DTConEx$ case. As the DropTail queue is entirely filled, the users experience the highest delay equal to 100ms. In $REDConEx$ and $FullConEx$, the queueing delay is reduced by the action of the RED queue. $REDConEx$ is reducing the queueing delay more than $FullConEx$ because the RED queue drops packets in the former case while it marks them in the latter. The congestion policer also contributes to reduce the queueing

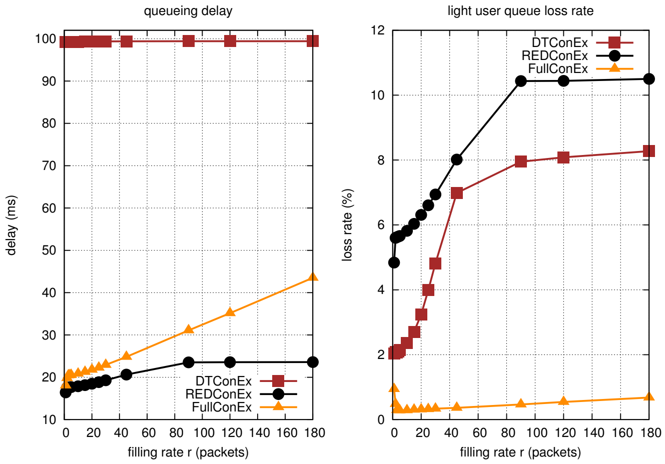


Figure 6. mean queueing delay and queue loss rate of a light user

delay which is further decreased as the filling rate decreases due to heavy users' policing.

By reducing traffic pressure on the bottleneck, the congestion policing also reduces the loss rate light users encounter, especially in $DTConEx$ and $REDConEx$, which are based only on losses in order to notify congestion. In both cases, the light users' loss rate decreases as the filling rate decreases. For all filling rates, $REDConEx$ shows a highest loss rate than $DTConEx$ because in the former case, the RED queue begins dropping packets earlier than the DropTail queue in the latter case. Finally, $FullConEx$, in which packets are ECN-marked rather dropped, shows a significantly lower loss rate for light users than the two other cases.

TABLE III. Performance summary

| Case | Fairness | Loss rate | Delay | Deployability |
|---|---|---|---|---|
| DTConEx | *** | ** | * | *** |
| REDConEx | * | * | *** | ** |
| FullConEx | ** | *** | ** | * |

Table III summarises the advantages and drawbacks of each case in terms of fairness improvement, loss rate, queueing delay and deployability.

## V. Short-Lived flows

Short-lived flows represent a great number of flows that cross the Internet (Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Web objects, etc.). These flows are just a few packets long, they finish during the slow-start phase (in few RTTs) before reaching their fair-share rate [15]. This section aims to see how ConEx, which is a closed-loop mechanism requiring a number of RTTs to gather congestion information, behaves with short-lived flows and if it does bring an improvement to the completion time of these flows.

For performance evaluation, we use the same topology as in Section IV but modify the traffic sources from saturated long-lived flows to short-lived flows lasting only 10 packets. We use the aggregated traffic model described in [16], which uses a gamma distribution for the flow inter-arrival time, with a newly generated flow every 6ms on average. The 10 heavy users will generate 80% of the flows while the light users will generate the remaining 20%. In order to experience congestion in the network, a Not-ConEx cross traffic of 90Mbps over the 100Mbps bottleneck is generated. A strict policer is used as described in Section III-F. We monitor the flow completion time as a performance metric.

Each simulation lasts 600s, 30 simulations are performed to obtain a single point with a 95% confidence interval which is depicted on the graphs.

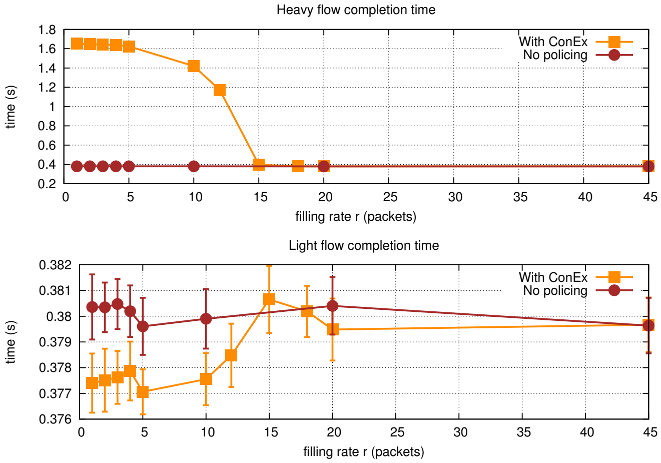


Figure 7. Completion time of a light and a heavy user's flow

Figure 7 represents the average completion time of a heavy user's flow and a light user's flow with and without the use of ConEx. The flows have an initial window of 2 segments and can be completed in 3 RTTs (300ms) which does not allow them to provide much congestion information for ConEx. Nevertheless, a heavy user can be policed when the filling rate is low enough ($r < 15 packets$), increasing greatly the completion time of his flows. The completion time of a 10-segment flow ranged from 380ms without policing up to 1.64s when policed. This is supposed to free the bottleneck for the light users' flows. Indeed, we see that when the heavy user is delayed, the light users benefit from a reduced completion time, but the decrease is only a few milliseconds, which is hardly a significant improvement.

Neither ConEx benefits from the use of short flows nor short flows benefit from ConEx. Short flows are not suited to retrieve congestion information for ConEx as they finish in few RTTs. These flows finish before they can react to policing. When short flows lose packets, they can see their completion time increase dramatically from a few milliseconds to several seconds because they might need to wait for an RTO to perform retransmissions and complete. As expected, ConEx behaves poorly in presence of short flows, and it should be even less interesting if, as [15] suggests, the initial window is increased to 10 segments, which represents a less favorable scenario than the simulated one. However, the poor behaviour of ConEx observed with short flows does not lessen the interest of the mechanism considering that long flows are the main source of congestion. If a per user congestion policer is used, it should be more profitable to focus on long flows, which can retrieve congestion information and can efficiently react to policing.

## VI. Video Streaming Traffic: YouTube use case

We have observed over the last years an impressive growth of the video streaming traffic in both Orange's fixed and mobile networks (36% for FTTH, 26% for Asymmetric Digital Subscriber Line (ADSL) and 39% for mobile downstream [1]). This led us to analyse how ConEx can alleviate the pressure caused by video streaming traffic and we chose as a use case the very popular YouTube plateform.

### A. *YouTube server model*

Many papers analysed the YouTube traffic generation. Among them, [17] [18] propose an algorithm to reproduce the behaviour of a YouTube server, which we implemented in NS2.

A server sends a video in two phases: the first phase is called the Initial Burst where 40s of video data is sent at maximum rate to provide sufficient buffering to the player. The second phase is called the Throttling phase where the server sends the rest of the video data in chunks with a $sending\ rate = 1.25 \times encoding\ rate$ of the video. The chunk size is 64KB and the chunks are sent over a TCP socket with a 2MB sending buffer.

### B. *YouTube player model*

We used the most precise monitoring approach proposed by [19] to implement a YouTube player in NS2. It is based on the status of the video buffer on the client player. The player starts playing the video when the buffered length exceeds a first threshold $\theta_0 = 2.2s$. If the buffer is depleted and the buffered length goes below a second threshold $\theta_1 = 0.4s$, the video stalls until the buffered length exceeds $\theta_0$, then the video can start anew. We retrieve from the video player the number of stalling events $N$ and their average length $L$ to compute the QoE following a model suggested by [20] with the following equation:

$$QoE(L, N) = 3.50\exp^{-(0.15L+0.19).N} + 1.50 \qquad (2)$$

### C. *YouTube results*

The same topology as in Section IV is used to perform the simulations with 10 heavy users and 50 light users. The simulated scenario is the following: in the first 100s of the simulation, the heavy users have 20 FTP flows downloading at the maximum rate they can reach. No light user is present yet, the 10 heavy users can equaly share the bottleneck. In the next 100s, the light users begin requesting, randomly and uniformly over the 100s, a video from the servers. This video has a 300s length and a bitrate of 1128kbps, which corresponds to the recommended bitrate for uploading 360p videos to YouTube (1000kbps for the video bitrate and 128kbps for the stereo audio bitrate [21]). The heavy users, which are responsible for 80% of the traffic, now have to share the network with the newcomers. At $t = 500s$, all light users should have finished watching their 300s video if no stalling events hampered the viewing, and the heavy users should be able to continue using the bottleneck until the end of the simulation 100s later. The mean QoE of the light users is computed at the end of each simulation.

A simple DropTail queue is used at the bottleneck. The policer is a strict policer as described in Section III-F and all users use cubic as a congestion control algorithm.

Figure 8 shows the throughput of the heavy and the light users versus time. The three time periods of the simulated scenario are shown: before the arrival of the light users (0s-100s), during the light users' presence (100s-500s), and after the presumed departure of the light users if they watched the videos smoothly (500s-600s).

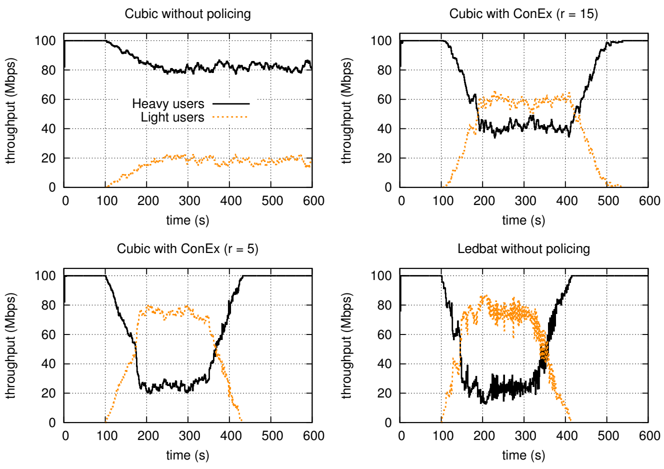


Figure 8. Throughput of heavy and light users versus time

Figure 9 represents the computed QoE, the number of stalling events and the duration of a single stalling event for a light user in the following three cases: using cubic as a congestion control algorithm for heavy users without policing, using cubic for heavy users with ConEx policing and using LEDBAT as a congestion control algorithm for heavy users without policing.

*a) Cubic without policing:* When no policer is used, TCP with cubic will share the bottleneck equally between flows. The heavy users get 80% of the bottleneck and the light users will not be able to watch the video before the end of the second period. The light users will still be active during the third period, reducing the throughput of the heavy users when compared to the first period. The light users see their video stall many times and for a long duration as shown in Figure 9, resulting in a $QoE = 1.5$, which is the lowest obtainable value with equation (2). Users with this low QoE would have stopped watching the video when the first stalling events occured.

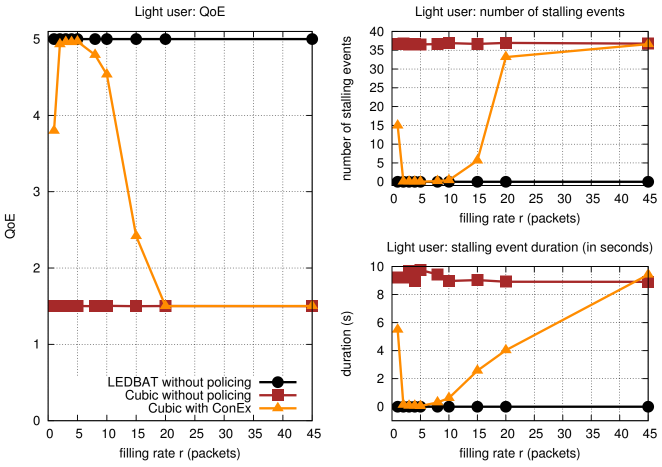


Figure 9. QoE of light users, the number of stalling events and the duration of a single stalling event

*b) Cubic with ConEx:* ConEx is activated in order to police the heavy users. Figure 9 shows that as the filling rate decreases, the light users' QoE increases to very good values

(QoE > 4) due to the reduced number of stalling events. The gain in QoE for the light users results from the heavy users decreasing their throughput, due to congestion policing, during the second period as represented in Figure 8. The light users are then able to finish viewing their video before the end of the second period. As the light users leave the bottleneck, the heavy users can increase their throughput during the third period.

*c) LEDBAT without policing:* The heavy users could avoid policing by being less aggressive towards video traffic. They could either postpone their activities until a less congested period, or they could use a less aggressive congestion control algorithm which yields the network ressources when encountering congestion. LEDBAT [22] is such a congestion control algorithm. It uses the available bandwidth in a bottleneck and yields in presence of standard TCP. When LEDBAT is used (implementated in NS2 by [23]) instead of cubic for the heavy users, results in Figure 8 show that, without requiring any policing, the heavy users decrease their throughput and the light users are able to watch their video with a very good QoE (Figure 9), similar to the results obtained by using cubic and ConEx policing ($r = 5$). When the light users' videos finish, LEDBAT is able to use the freed resources in the bottleneck.

As suggested in the ConEx charter [2], ConEx can be deployed in order to incentivize the heavy users to use a LEDBAT-like congestion control mechanism. The use of LEDBAT prevented the heavy users from consuming tokens for applications like file transfer, preserving their congestion allowance for more critical applications, while allowing the light users to have a good quality of experience. If the video delivery relies on HTTP-adaptive streaming, the light users would decrease the resolution of their video when encountering congestion, but after the heavy users have reduced their throughput using LEDBAT or in response to ConEx policing, the light users could increase the resolution of their video and benefit from a higher video quality.

## VII. Summary and conclusion

ConEx is a new mechanism that allows a user to inform the network of the amount of congestion encountered. This allows the network operator to implement congestion-based policies proportionally to the amount of congestion a user has contributed to.

In Section IV, we have seen that ConEx allows us to differentiate between a light and a heavy user to improve the fairness between users. We have shown that ConEx can still improve fairness even with the minimum modifications (the ability to react to lost packets by sending a Re-Echo-Loss signal) and the use of simple DropTail queues. So, an efficient initial deployment is possible, as suggests [3], before considering the deployement of a more accurate ConEx relying on ECN, which requires modifications to both the senders and the receivers and the use of RED queues. The advantages and drawbacks of each step of modifications are summarized in Table III.

In Section V, we argued that neither ConEx benefits from the use of short flows nor short flows benefit from ConEx. Indeed, the short flows do not provide enough congestion information to ConEx, and policing them is not beneficial for their completion time. It is more profitable to focus on policing long and responsive flows.

In Section VI, we have seen how video streaming like YouTube can benefit from ConEx. ConEx can be used to restrain the heavy users who do not yield voluntarily under congestion, while leaving unpoliced those who do through a congestion control mechanism like LEDBAT. This should provide incentives for the heavy users to be more cooperative during congestion periods. The use of LEDBAT can protect the heavy users from being policed through ConEx while allowing the light users to have a great QoE.

## VIII. Future Work

Implementing a per user congestion policer requires the determination of the policer's parameters, the filling rate (the allowed Congestion-Rate) and the bucket depth (the allowed Congestion-Burst). Different kinds of flows with different behaviours need to be policed with the same allowance rate which makes the determination of these parameters challenging. Further studies are required on this subject.

The congestion policing function is the key to improve fairness between users and to enforce some users to yield if they do not voluntarily. Designing a policer algorithm that achieves the goals we set is a crucial point in the deployement and is one of the main objectives of our future work.

Finally, the auditor can be necessary if there is a risk that the sources do not report the right Congestion-Volume they encounter. If auditing is relatively easy when ECN is used, ConEx on loss is more challenging as it requires detecting lost packets in the auditor. To address these issues, we can harness the substantial work concerning the auditor that has been done under the Trilogy project [5].

## References

[1] M. Feknous, T. Houdoin, B. Le Guyader, J. De Biasio, A. Gravey, and J. Torrijos Gijon, "Internet traffic analysis: A case study from two major european operators," in Computers and Communication (ISCC), 2014 IEEE Symposium on, June 2014, pp. 1–7.

[2] ConEx Working Group Charter. [Online]. Available: http://datatracker.ietf.org/wg/conex/charter/ [retrieved: March, 2015]

[3] B. Briscoe, R. Woundy, and A. Cooper, "Congestion exposure (conex) concepts and use cases," December 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6789.txt [retrieved: March, 2015]

[4] M. Kühlewind and R. Scheffenegger, "Tcp modifications for congestion exposure," November 2014. [Online]. Available: http://www.ietf.org/id/draft-ietf-conex-tcp-modifications-07.txt [retrieved: March, 2015]

[5] B. Briscoe et al., "Final report on resource control, including implementation report on prototype and evaluation of algorithms," December 2010. [Online]. Available: http://www.trilogy-project.org/fileadmin/publications/Deliverables/D13_-_Final_report_on_resource_control__including_implementation_report_on_prototype_and_evaluation_of_algorithms.pdf [retrieved: March, 2015]

[6] M. Kühlewind and M. Scharf, "Implementation and performance evaluation of the re-ecn protocol," in Incentives, Overlays, and Economic Traffic Control, ser. Lecture Notes in Computer Science, B. Stiller, T. Hoßfeld, and G. Stamoulis, Eds. Springer Berlin Heidelberg, 2010, vol. 6236, pp. 39–50. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15485-0_5 [retrieved: March, 2015]

[7] F. Mir, D. Kutscher, and M. Brunner, "Congestion exposure in mobility scenarios," in Next Generation Internet (NGI), 2011 7th EURO-NGI Conference on, June 2011, pp. 1–8.

[8] Y. Zhang, I. Johansson, H. Green, and M. Tatipamula, "Metering re-ecn: Performance evaluation and its applicability in cellular networks," in Teletraffic Congress (ITC), 2011 23rd International, Sept 2011, pp. 246–253.

[9] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” IEEE/ACM Trans. Netw., vol. 1, no. 4, pp. 397–413, Aug. 1993. [Online]. Available: http://dx.doi.org/10.1109/90.251892 [retrieved: March, 2015]

[10] K. Ramakrishnan, S. Floyd, and D. Black, “The addition of explicit congestion notification (ecn) to ip,” September 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3168.txt [retrieved: March, 2015]

[11] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, “Data center tcp (dctcp),” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, pp. –, Aug. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=2043164.1851192 [retrieved: March, 2015]

[12] M. Kühlewind and R. Scheffenegger, “Design and evaluation of schemes for more accurate ecn feedback,” in Communications (ICC), 2012 IEEE International Conference on, June 2012, pp. 6937–6941.

[13] The Network Simulator - ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/ [retrieved: March, 2015]

[14] A. Martin and M. Menth, “Conex-based congestion policing – first performance results,” March 2012. [Online]. Available: http://www.ietf.org/proceedings/83/slides/slides-83-conex-5.pdf [retrieved: March, 2015]

[15] N. Dukkipati et al., “An argument for increasing tcp’s initial congestion window,” SIGCOMM Comput. Commun. Rev., vol. 40, no. 3, June, 2010, pp. 26–33. [Online]. Available: http://doi.acm.org/10.1145/1823844.1823848 [retrieved: March, 2015]

[16] S. Gebert, R. Pries, D. Schlosser, and K. Heck, “Internet access traffic measurement and analysis,” in Proceedings of the 4th International Conference on Traffic Monitoring and Analysis, ser. TMA’12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 29–42. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-28534-9_3 [retrieved: March, 2015]

[17] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. Lopez-Soler, “Analysis and modelling of youtube traffic,” Transactions on Emerging Telecommunications Technologies, vol. 23, no. 4, June, 2012, pp. 360–377. [Online]. Available: http://dx.doi.org/10.1002/ett.2546 [retrieved: March, 2015]

[18] J. Ramos-munoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. Lopez-soler, “Characteristics of mobile youtube traffic,” Wireless Communications, IEEE, vol. 21, no. 1, February, 2014, pp. 18–25.

[19] R. Schatz, T. Hossfeld, and P. Casas, “Passive youtube qoe monitoring for isps,” in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, July 2012, pp. 358–364.

[20] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, “Internet video delivery in youtube: From traffic measurements to quality of experience,” in Data Traffic Monitoring and Analysis, ser. Lecture Notes in Computer Science, E. Biersack, C. Callegari, and M. Matijasevic, Eds. Springer Berlin Heidelberg, 2013, vol. 7754, pp. 264–301. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36784-7_11 [retrieved: March, 2015]

[21] Google, “Advanced encoding settings.” [Online]. Available: https://support.google.com/youtube/answer/1722171 [retrieved: March, 2015]

[22] S. Shalunov, G. Hazel, J. Iyengar, and M. Kühlewind, “Low extra delay background transport (LEDBAT),” December 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6817.txt [retrieved: March, 2015]

[23] D. Rossi, C. Testa, S. Valenti, and L. Muscariello, “Ledbat: The new bittorrent congestion control protocol,” in Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on, Aug 2010, pp. 1–6.

# Toward a semantic based signage digital system: Mediactif

Caroline Collet
PERTIMM
Asnières, France
Email: caroline.collet@pertimm.com

Corinne Berland
ESIEE-Paris
ISYS Department
Noisy le Grand, France
Email: corinne.berland@esiee.fr

Samuel Ben Hamou
Thierry Simonnet
ESIEE-Paris
R&D Department
Noisy le Grand, France
Email: {samuel.benhamou, thierry.simonnet}@esiee.fr

***Abstract*—Digital signage systems distribute all kinds of information to specific locations, including retail stores, public areas and transportations. Thanks to the massive adoption of displays and wide application of wireless networks, digital signage can deliver targeted messages designed to accurately reach the passing audience and eventually influence customers. The aim of the MEDIACTIF project is to develop a new digital signage concept, using both human traffic modelling and compartmental behavior based on interests. The major issue we are facing as of now is that crowd motion has extensively been studied on particuliar test cases, on both macroscopic and microscopic levels, but it is lacking considerable social and personal inputs, which are significant for each site and scenario.**

***Keywords*–*digital signage; real-time; network; sensors.***

## I. INTRODUCTION

The MEDIACTIF Project [1] has been started a year ago to propose solutions to different problems:

- Reducing the stress of fair visitors or airports clients by reducing crowd congestions and increasing the pedestrian flows.
- Giving relevant information to users and the operations team in real time.
- Offering a centralized system that may adapt signage to any situation.
- Increase security levels with adaptive signage (redirect pedestrians easily in case of emergency situations)
- Reducing global waste of both printed signage materials, but also aiming at a massive drop in the consumption of high toxicity materials like inks.

Partners on this project have different fields of expertise and some of them are facing day to day issues regarding their fixed or dynamic signage impact over people behaviors. They would like to efficiently inform clients or simply redirect them to a specific location.

The MEDIACTIF Project aims not only to display different contents on screens, but also to integrate multiple sensors in the processing loop. This would allow for signage adaptation depending, for example, on:

- crowd densities
- personal preferences
- commercial factors of interest
- specific or recurrent events
- previously computed models and statistics

These models should be driven by the analysis of previous sessions (when available) and will be enhanced with the sensors live input. To reach this aim, Mediactif system uses different kinds of sensors, a model engine and semantic tools.

## II. SYSTEM ARCHITECTURE

MEDIACTIF will be an adaptive and centralized system. All relevant data will first be collected by a specific module and will serve to compute crowd models. These will be used for crowd traffic prediction. They will act as the base module for all corporate models like airports and event organizers. It should also allow mobile services (free or not) for visitors via apps or web services. According to the state of the art regulations, it has to follow network standards to pilot existing digital signage systems and to communicate with its own devices. The back-office architecture is organized around communication, events and sensors management. For a fair, or for an airport terminal, it is possible to define basic events or rules (holidays, week-end, plane arrivals and departures) and associate, to each of them, fixed plans to anticipate crowd variations. Using key sensors values, it is possible to manage adaptive modes (last minute gate changes, system failure, human weaknesses, etc.). One of the key rules will be to immediately handle a crowd increase, but awaiting stabilization for a decrease.

## III. SENSORS

Sensors, in MEDIACTIF, are to give relevant information concerning crowd observation. When dealing with human sensing, we can enumerate five different objectives: presence detection, counting, location, tracking an identification [2]. In the case of non instrumented people, for presence detection, simple binary sensors can be used : e.g., passive infrared (PIR), pressure sensitive tiles, electric field sensors and vibration sensors. Among the limitations of these sensors, we can highlight for the PIR sensor the fact that it can't detect an immobile person. Pressure sensitive tiles can't enable the distinction of two close persons (50 cm)[3] and vibration sensors are very sensitive to noise. Other non instrumented sensors, based on signal propagation, can also be used for our project: radio signals, acoustic wave, laser. More precisely, Radar, sonar and

ISBN: 978-1-61208-404-6

ladar are used, according to either the attenuation of the signal or the propagation time of the signal, to reconstruct an image. Ultra-wide Band (UWB) can be used for detection and localization [4], and in the case of radio propagation, the measure of the doppler effect is a mean to obtain movement detection. Camera is, of course, an appropriate solution for detection, identification, tracking, counting and people queue estimation, with appropriate signal processing algorithms [5][6]. For our project, the instrumented approaches we can use are either a device to device approach, with specific application developed on mobile phones, or the scanning of radio signals in the environment, such as WIFI signal [7]. In the case of device to device approach, the localization can be as precise as 20 cm with time of arrival (TOA) and time difference of arrival (TDOA) principles [8]. In the context of MEDIACTIF, the choose of the sensors will depend on the possibility to use instrumented or non instrumented systems. The more appropriate instrumented solution is the use of localization based on TOA/TDOA, coupled with area sensing. If we have to choose non instrumented solutions, the best solution would be the use of camera, coupled with dopplers sensors. In term of price, the solution with the lowest cost is the use of binary sensors.

## IV. Crowd analysis and models

The major issue in handling pedestrians is to define accurate models valid for a wide range of topologies and flow densities. Models should also be realistic, robust against incomplete data, and of course computationally manageable. Pedestrian dynamics share some similarities with fluids, and it is not surprising that the first models of crowds were inspired by hydrodynamics or kinetics of gases. The main idea is to consider that the movements of a pedestrian in the crowd are similar to movements of a particle in a gas. Based on this assumption, it is possible to make use of tools from Newtonian mechanics to describe the behavior of a pedestrian by means of attractive and repulsive forces. The pedestrian is attracted toward a destination point, but at the same time repulsed by other pedestrians. Henderson found as early as 1971, from measurements of motion in crowds, a good agreement of the velocity distribution functions with Maxwell-Boltzmann distribution [9]. Social forces have been introduced by Helbing in a microscopic model [10] based on the idea that pedestrians have different perceptions about intimate/personal and social space, which leads to repulsive forces between persons. Cellular automata [11], [12], [13], [14] are another important class of models that are discrete in space and time. Most of these models represent pedestrians by particles that can move to one of the neighboring cells based on transition probabilities which are determined by the desired direction of motion, interactions with other pedestrians, and interactions with the infrastructure (walls, doors, etc.).

### A. Rejected models

However, unlike Newtonian particles, persons have a free will and may want to avoid jams by giving up their preferred path when approaching a crowded area to find a new path. To take into account such strategies, microscopic models are to be extended well above the present state of the art. Stochastic behavioral rules may lead to potential realistic representations of complex systems like pedestrian crowds, however, parameterization and calibration of such models may remain elusive.

The aim of the MEDIACTIF project is to have an enhanced digital signage system. This implies that all signage could be updated depending on crowd behavior, external events... The key point is modeling the crowd behavior, using some key rules. That's why some models are not relevant for the project purposes:

- Particle mechanics based models : all pedestrians interact between themselves as particles. This model subset uses a collection of global rules. The problem is that pedestrians are not particles. To describe precisely the movements of a pedestrian with Newtonian forces, one usually needs pretty sophisticated equations of motion, which are hard to calibrate. Moreover, the movements of pedestrians during computer simulations look pretty artificial and sometimes obviously not realistic. It is impossible to direct individual entities when necessary.
- Fluid mechanics based models : at the beginning of '70s, Henderson worked on a fluid mechanics based modelization [15]. It was a global crowd modelization, with one set of rules for all particles.
- Cellular automaton models : the simpliest way of cellular automation is the deterministic model of Fukui-Ishibash.
- Predator-Prey models : crowd behavior could be modelized using Lotka-Volterra equation [16].
- Epidemiological models : some crowd events could uses this kind of rules. But, if it can be used for disease dissimination using communication means (plane, train, road etc.) it is far less useful for a commercial center or a fair.

In fact, any global equation based models doesn't fit because it is mandatory to distribute some comportemental rules to a set of pedestrian (like family, businessman, elderly people...). And then we decided to select an agent based model [17].

### B. Tools

MEDIACTIF project implements 3 main functional elements needed for a crowd characterization : crowd analysis from sensors, model definition and use for prediction, and off-line simulation.

### C. Crowd Analysis

The first step to determine crowd behavior is to analyze its comportment in typical situations. Criteria choice is the most important thing to refine. In fact, these criteria must be the same for analysis but also for the model definition and of course for simulation.

To handle digital signage impact, it is necessary to consider the vision field of each pedestrian. If provided information is walking time to a specific place (20 minutes by the left way, 5 by the right one), its possible impact is only on pedestrians that have this information in line of sight. Depending on sensors, it is necessary to exploit every solution in Vision, Learning and Pattern Recognition to select crowd-scene key behaviors.

Some Smart Cities projects try to implement these kind of functionalites. For example, the Inria Project Lab City-Lab@Inria is currently under creation and studies information and communications technology (ICT) solutions to promote

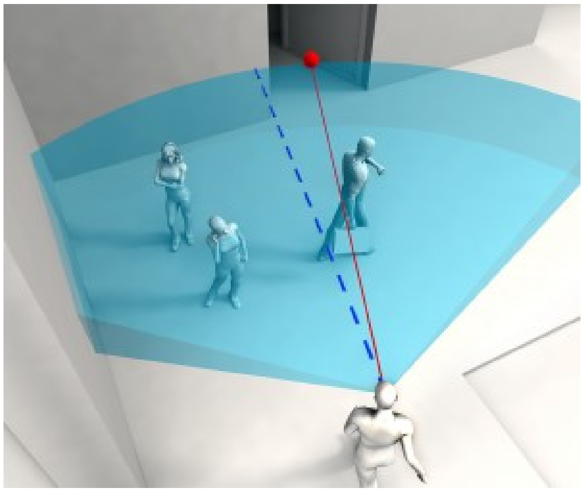


Figure 1. Vision Field Model

social and environmental sustainability and facilitate the transition to Smart Cities. Another example, CNR-ISTI in Italy: HIIS Lab develops a framework for improving the multi-device user experience in Smart Cities. This framework proposes a solution that extends existing Service Front Ends for Smart Cities and supports multi-device interaction by exploiting personal devices and public displays. It can be used to easily obtain interactive multi-device applications for different domains and in different contexts in Smart Cities including single or multi-user applications, indoor or outdoor environments and mobile and stationary devices. ([18][19])

### D. Multi-agent Model

There are different kind of models, with their own algorithms [20] [21]. To manage the crowd as a group of individuals with their own interests, their own objectives, their physical characteristics, only the multi-agent models, individual-based, can match the needs of the project, with very diverse characteristics (airports, trade shows, fairs ...). A good model should be able to predict the emergence of known collective crowd behavior, such as lane formation or crowd turbulence.

The proposed theoretical model is based on a fairly simple concept [22]. The first step is to mathematically describe the visual information that a pedestrian can see in a crowd and how a pedestrian behaves in a crowd. For this, it is necessary to define the limits of the visual field and calculate the distance before colliding with other individuals and obstacles (see Figure 1).

The next step is to set two simple rules to adjust the trajectory of the pedestrian, especially its speed and direction based on the previously calculated visual information.

- The first one is to choose a direction of travel that minimizes coverage of vision without deviating too much from the point of destination.
- The second one is to set walking speed to maintain a safe distance to the nearest obstacle or individual.

These rules are called heuristics - a term describing cognitive science quick decisions that people do not think too much about their behavior.

The last step is to realistically reproduce a moving crowd in congestion situations, the movement rules are combined with the physical forces that occur during unintentional body contact.

### E. Simulation

Crowd simulation is necessary to setup properly sensors and displays to provide an efficient sytem based on dynamic signage. Modelling is the basic tool for simulation. Numerical simulations of the model previously described is able to generate a large variety of collective behaviors, such as the spontaneous separation of opposite flows of pedestrians in bidirectional traffic. The model can also predict typical situation as bottleneck or congestion due to, e.g., a right angle corridor. The results are promising and seem far better than other simulators we evaluated. The evaluated tools integrate behavioral functions. They simulate an average behavior if an peculiar event occurs. But none of them integrates a behavior action when a signage change (for example : going left instead of going right if a congestion accurs):

- Anylogic [23]: used by many airports. AnyLogic provides complex solutions to plan, manage, and optimize pedestrian flows in public buildings like airports, railway stations, shopping malls, and stadiums. The software can evaluate the capacity of buildings and certain objects, find and avoid pedestrian bottlenecks, optimize business processes at service points, carry out evacuation planning, assess shopping area customer traffic, evaluate parking, roadway, and public transportation accessibility. But it is difficult to integrate vision in a model.
- SimWalk [24]: integrate some tools for external events like baggage, interconnection but no vision parameters nor signage simulation.
- Pedestrian dynamics [25]: Adaptation to dynamically changing (local) conditions. As in reality the situation can change during the simulation.
- Mass Motion [26]: This system is highly scalable for large crowds and for simulations scenarios that cover multiple days.
- CAST Pedestrian [27]: airport facilities solution but simulation doesn't handle external events.
- MATSim + Via [28]: The mobility data provided by Senozon, or the results of MATSim simulations in general, can be very large and do not give much away in the raw format. The analysis tool allows to easily extract important characteristics of the data in a short time. It is useful to extract main characteristic of a flow but is less relevant for crowd.
- Pedsim [29]: this simulator is in fact a library to create a specific simulation tool. There are many ”odd” comportment especially for right angle congestion .
- PETrack : pedestrian trajectory extracting tool using a video stream.
- JuPedSim : Jlich Pedestrian Simulator is an open source framework for simulating pedestrian dynamics. Several well-known models from the literature are implemented. It provides researchers and students preparing their projects an appropriate environment to

model and simulate pedestrian dynamics. This product is at an early stage and tests don't give better result the particle or fluid models.

To evaluate and predict behavior when using MEDIACTIF system, it is necessary to develop a simulator that handle characteristics of dynamic signage. Otherwise, there will be a simulation and prediction tool that will not give relevant results; as if for a crowd of rabbits using grass resource of a field, simulation use goat comportment parameters.There will be no chance to have results that are valid, consistent or usable.

## V. SEMANTICS

### A. Related works

*1) Semantic inside signage digital systems:* The display of contents adapted to users recently became a very attractive research field especially for business purposes. With the democratization of mobile phones and a continuous trend towards big data, particularly oriented on users preferences and inclinations, it is now easy to display personalized content [30]. One of the current objectives nowadays is to suggest places [31][32] or offers [33] best suited to each user. With the smartphone applications ecosystems, it is now relatively easy to get users' information and then use state of the art methods like collaborative filtering [32] or model based recommendation methods [31].

The challenge now is to propose the same recommendation efficiently on common displays in generic areas while over the years 2000, researchers focused only on screens in specific contexts like for instance offices [34] or inside sport halls [35], etc.

Another difficulty arises in taking into account groups and not only single users as primary targets: past recommendation algorithms were well adapted to single users but are falling short for groups. [34], [36] and [35] attempted to solve this shortcoming using very simple methods like averaging each users' scores.

Thus, making precise recommendations to a group of users still proves to be quite challenging. Moreover, attempting to make these on regular displays and not only on mobiles is adding an extra layer of complexity since it becomes more complicated to obtain individual inputs, not even accounting for the actual up difficulty to keep abreast of the content to display [37]. Search engines, are not only good alternatives to data mining methods since they remove sparsity issues, but also allow us to easily add, update and delete content in real time with little to no impact on the searches. Therefore, in this project, we attempted to use a search engine inside a digital signage system to drive content suggestions for single users as well as groups.

*2) Pedestrian navigation on display screens:* The MEDIACTIF project aims partly at helping pedestrians to navigate correctly both indoor (inside buildings [38], for demonstrations, conferences...) and outdoor (in towns, subways [39]...) [40]. Usually, mobile phone applications are used for this particular job. The obvious advantage is that we are able to get information such as a destination by simply asking a user, and then to compute the most appropriate path for him to get there. It is still slightly basic and user centric, thus implicating multiple display screens seems like a natural evolution. When a user passes in front of a display, his mobile phone can share his destination with the display. Then the display can show an updated map with proper redirections calculated in real time [41] and corresponding to the present geographical location [42]. Some researchers are even trying to mix the previous methods of collaborative filtering in order to guide users to their destinations and potential Points of Interest (POI) [43].

This is precisely what we are focusing on in the MEDIACTIF project : to allow for some people flow regulations.

### B. Pertimm

Pertimm is providing a search engine to major e-commerce customers. The idea here is to integrate Pertimm's search engine inside the global system in order to not only regulate people flows but also to suggest activities corresponding to people desires and of course, having the possibility to select points of interest suited to both a group of person as well as single users. Therefore, the development is following a three steps approach: We will first create a basic system of search recommendation only based on simple queries in order to integrate Pertimm inside the system. Then, we will expand the functionalities with content based recommendations. Finally, we will upgrade the previous solution in order to create a system based on collaborative filtering for multiple users.

*1) First step : basic search.:* To enable the system to use the search engine as a basic search, it is necessary to first use devices that enable users to make search requests. Indeed, with absolutely no knowledge about the user, it is not possible for us to generate any appropriate request/answer. Thus it is necessary to let the user drive the system. This is actually how most digital signage systems currently work. Let us picture a user in a mall, looking for a particular shop on a store locator device (usually some big table/screen). He has the ability to make keyword requests about shops to look for either by brand, categories, interests... Using MEDIACTIF, we would not want to rely only on this particular vision. The idea is to make automatically pre-recorded requests to display suitable contents on screens even if the user has not shown any particular interest so far.

Explanation of the first experiment (vestiaires)

*2) Second step : content based recommendation.:* Content based recommendation is a method that only uses past users interests in order to propose new points of interest[44][45]. The idea is to extract characteristics of ones' history in order to match characteristics of the new suggestions. On this particular part, Points of Interest (POI) have to be described usually with natural language and keywords (that can be figures). Then, selecting them usually involves a keyword matching algorithm, sometimes combined with a frequency method.

To allow this, we then need to know users previous interests. Thus, we plan to make users fill a form at a particular time (e.g., when they buy their tickets for an exposition). Each interest on the form will then be indexed and the suggestions will arise from a confrontation with the global points of interest index (stands, restaurants...with their description).

When one user needs a suggestion, we will look for a relevant characteristic inside all his interests and we will query the search engine to find one or multiple matching POIs. Though it is perfectly suited to mobile phones, with regular display screens, people will be rarely standing alone in front of it. Therefore, the search engine has to deal with several users.

The idea is to consider these users as a single one. We plan to look for their main shared interest characteristics and use it as a query token.

With this kind of recommendation, we would eventually suggest to foreign people a typical french restaurant in the area if most of them like french cuisine. The major drawback here is that we need users to be compliant and fill in a form.

*3) Third step : collaborative filtering.:* The objective is to suppress any user implication inside the system by implementing pure collaborative filtering. Collaborative filtering is currently the most used recommendation method. It works by saving every users points of interests inside a matrix [46][47] and creating similarity matrices by comparing each user one by one. Once a score of similarity is computed, suggestions to one user coming from other similar users can be given.

We plan on using Pertimm search engine to simulate this process: we will index every user with its POIs and then use search engine methods like catalogs to generate recommendations. Using a search engine will allow us to avoid the calculations and storage of big matrices.

Still to obtain users POIs, we need to be able to follow them and store their behaviors. We plan to do so with a phone application, storing queries and locations.

For the group perspective, we will create a dummy user with POIs driven from the majority of the users as will suggestions be. As a result, we will be able to combine collaborative filtering with previous content based method to obtain recommendation [48][49].

## VI. Conclusion

The MEDIACTIF Project is at its very early stages of development. Surely some critical points must be solved first. Models must be setup to take care of the different usage scenarios considered so far (fairs, airports, etc.). We must handle people behaviors depending on tangible parameters (time, events, etc.) but also on unquantifiable parameters like personal interests. Another point to solve is indoor localization. A multimodal approach is planned using physical sensors and networking (Wi-Fi) though we are also looking at other options based on mesh networks.

## Acknowledgment

ESIEE-Paris works in close cooperation with !nstantan (project leader), Aéroports de Paris, VIPARIS, Scilab Enterprises, INNES, PERTIMM, B2B EN-TRADE. Some research leading to these results has received funds from the French Governement (DGCIS), Conseil Régional d'île de France and Conseil Général de Seine Saint-Denis for the FUI 16 program.

## References

[1] "Mediactif Project," "http://mediactif.esiee.fr/", 2014, [Online; accessed 28-Fev-2015].

[2] T. Teixeira, G. Dublon, and A. Savvides, "A survey of human-sensing: Methods for detecting presence, count, location, track, and identity," in ENALAB technical report, 2010.

[3] T. Murakita, T. Ikeda, and H. Ishiguro, "Human tracking using floor sensors based on the markov chain monte carlo method," in Proceedings of the 17th International Conference on. ICPR 2004, 2004, pp. 917–920.

[4] R. Zetik, S. Crabbe, J. Krajnak, P. Peyerl, J. Sachs, and R. Thomä, "Detection and localization of persons behind obstacles using m-sequence through-the-wall radar," vol. 6201, 2006, pp. 62 010I–62 010I–12.

[5] V. Parameswaran, V. Shet, and V. Ramesh, "Design and validation of a system for people queue statistics estimation," in Video Analytics for Business Intelligence, ser. Studies in Computational Intelligence, C. Shan, F. Porikli, T. Xiang, and S. Gong, Eds. Springer Berlin Heidelberg, 2012, vol. 409, pp. 355–373.

[6] M. Isard and A. Blake, "Condensation: conditional density propagation for visual tracking," in International journal of computer vision. ICPR 2004, 1998, pp. 5–28.

[7] Z. Junyang, W.-C. Yeung, and J.-Y. Ng, "Enhancing indoor positioning accuracy by utilizing signals from both the mobile phone network and the wireless local area network," in Advanced Information Networking and Applications, 2008. AINA 2008. 22nd International Conference on, 2008, pp. 138,145.

[8] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha, "Tracking moving devices with the cricket location system," in In Proceedings of the 2nd ACM International Conference on Mobile Systems, Applications, and Services (MobiSys04. ACM Press, 2004, pp. 190–202.

[9] L. Henderson, "The statistics of crowd uids," vol. 229. Nature, 1971, pp. 381–383.

[10] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," vol. 51. Phys. Rev E, 1995, pp. 4282–4286.

[11] V. Blue and J. Adler, "Cellular automata microsimulation of bi-directional pedestrian ows," vol. 1678. Transportation Research Board, 1999, pp. 135–141.

[12] M. Fukui and Y. Ishibashi, "Jamming transition in cellular automaton models for pedestrians on passageway," vol. 68. Journal of the Physical Society of Japan, 1999, pp. 3738–3739.

[13] ——, "Self-organized phase transitions in cellular automaton models for pedestrians," vol. 68. Journal of the Physical Society of Japan, 1999, pp. 2861–2863.

[14] P. G. Gipps and B. Marksj´o, "A micro-simulation model for pedestrian ows," vol. 27. Mathematics and Computers in Simulation, 1985, pp. 95–105.

[15] L. Henderson, "A micro-simulation model for pedestrian ows," vol. 8. Transportation Research, 1974, pp. 509–515.

[16] S. Kim, C. Hoffmann, and J. M. Lee, "An experimental in rule-based crowd behavior for intelligent games." in AISS, vol. 2, no. 3, 2010, pp. 32–39.

[17] H. Van Dyke Parunak, R. Savit, and R. L. Riolo, "Agent-based modeling vs. equation-based modeling: A case study and users guide," in Multi-agent systems and agent-based simulation. Springer Berlin Heidelberg, 1998, pp. 10–25.

[18] J. Müller, F. Alt, A. Schmidt, and D. Michelis, "Requirements and design space for interactive public displays," in ACM Multimedia, 2010, pp. 1285–1294.

[19] S. Hosio and et al., "Supporting distributed private and public user interfaces in urban environments," in proc. of HotMobile, 2010, pp. 25–30.

[20] V. D. Phung, A. Drogoul, and N. D. Nguyen, "modèles dynamique de populations: implémentation des modèles mathématiques et informatique dans gama." Institut de la Francophonie pour lÍnformatique Hanoi, 2009.

[21] A. Drogoul, C. T. Quang, and E. Amouroux, "Agent-based simulation: definition, applications and perspectives," 2008.

[22] M. Moussad, D. Helbing, and G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters," in Proceedings of the National Academy of Sciences, vol. 108, no. 17, 2011, pp. 6884–6888.

[23] "Anylogic," "http://www.anylogic.fr/", 2014, [Online; accessed 28-Fev-2015].

[24] "Simwalk," "http://www.simwalk.com/", 2014, [Online; accessed 28-Fev-2015].

[25] "Pedestrian Dynamics," "http://www.pedestrian-dynamics.com/", 2014, [Online; accessed 28-Fev-2015].

[26] "Mass Motion," "http://www.oasys-software.com/", 2014, [Online; accessed 28-Fev-2015].

[27] "CAST Pedestrian," "http://www.airport-consultants.com/", 2014, [Online; accessed 28-Fev-2015].

[28] "MATSim + Via," "http://www.senozon.com/, 2014, [Online; accessed 28-Fev-2015].

[29] "Pedsim," "http://pedsim.silmaril.org/, 2014, [Online; accessed 28-Fev-2015].

[30] J. Müller, A. Krüger, and T. Kuflik, "Maximizing the utility of situated public displays." in User Modeling, C. Conati, K. F. McCoy, and G. Paliouras, Eds., vol. 4511. Springer, 2007, pp. 395–399.

[31] M.-H. Park, J.-H. Hong, and S.-B. Cho, "Location-based recommendation system using bayesian user's preference model in mobile devices." in UIC, J. Indulska, J. Ma, L. T. Yang, T. Ungerer, and J. Cao, Eds., vol. 4611. Springer, 2007, pp. 1130–1139.

[32] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments." in SAINT. IEEE Computer Society, 2006, pp. 124–129.

[33] S. Yuan and Y. W. Tsao, "A recommendation mechanism for contextualized mobile advertising," Expert Syst. Appl., vol. 24, no. 4, 2003, pp. 399–414.

[34] J. F. McCarthy, T. J. Costa, and E. S. Liongosari, "Unicast, outcast & groupcast: Three steps toward ubiquitous, peripheral displays." in Ubicomp, ser. Lecture Notes in Computer Science, G. D. Abowd, B. Brumitt, and S. A. Shafer, Eds., vol. 2201. Springer, 2001, pp. 332–345.

[35] J. F. McCarthy and T. D. Anagnost, "Musicfx: an arbiter of group preferences for computer supported collaborative workouts." in CSCW, W. A. Kellogg and S. Whittaker, Eds. ACM, 2000, p. 348.

[36] T. R. Payne, E. David, N. R. Jennings, and M. Sharifi, "Auction mechanisms for efficient advertisement selection on public displays," in EUMAS, B. Dunin-Keplicz, A. Omicini, and J. A. Padget, Eds., vol. 223. CEUR-WS.org, 2006.

[37] F. R. S. G. Ribeiro and R. Jos, "Timeliness for dynamic source selection in situated public displays." in WEBIST. INSTICC Press, 2009, pp. 667–672.

[38] F. Lyardet, D. W. Szeto, and E. Aitenbichler, "Context-aware indoor navigation," in Ambient Intelligence. Springer, 2008, pp. 290–307.

[39] M. Arikawa, S. Konomi, and K. Ohnishi, "Navitime: Supporting pedestrian navigation in the real world," Pervasive Computing, IEEE, vol. 6, no. 3, 2007, pp. 21–29.

[40] M. Kourogi, N. Sakata, T. Okuma, and T. Kurata, "Indoor/outdoor pedestrian navigation with an embedded gps/rfid/self-contained sensor system," in Advances in Artificial Reality and Tele-Existence. Springer, 2006, pp. 1310–1321.

[41] E. Rukzio, M. Müller, and R. Hardy, "Design, implementation and evaluation of a novel public display for pedestrian navigation: the rotating compass," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2009, pp. 113–122.

[42] J. Müller, M. Jentsch, C. Kray, and A. Krüger, "Exploring factors that influence the combined use of mobile devices and public displays for pedestrian navigation," in Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges. ACM, 2008, pp. 308–317.

[43] H. Huang and G. Gartner, "Using context-aware collaborative filtering for poi recommendations in mobile guides." in Advances in Location-Based Services. Springer, 2012, pp. 131–148.

[44] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," Communications of the ACM, vol. 40, no. 3, 1997, pp. 66–72.

[45] C. Basu, H. Hirsh, W. Cohen et al., "Recommendation as classification: Using social and content-based information in recommendation," in AAAI/IAAI, 1998, pp. 714–720.

[46] D. Billsus and M. J. Pazzani, "Learning collaborative information filters." in ICML, vol. 98, 1998, pp. 46–54.

[47] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in Proceedings of the 10th international conference on World Wide Web. ACM, 2001, pp. 285–295.

[48] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in AAAI/IAAI, 2002, pp. 187–192.

[49] I. Soboroff and C. Nicholas, "Combining content and collaboration in text filtering," in Proceedings of the IJCAI, vol. 99, 1999, pp. 86–91.

# A Self-contained Software Suite for Post-Disaster ICT Environment Using Linux Live USB

Vaibhav Garg, Kotaro Kataoka, Sahil Sachdeva

Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email: {cs12m1014, kotaro, cs11b032}@iith.ac.in

***Abstract*—This paper proposes a self-contained software suite for post-disaster situations packaged in Linux Live USB flash drive, that enables ad-hoc Information and Communication Technology (ICT) environment. In post-disaster situations, recovery of ICT environment including network and user devices takes certain amount of time. Moreover, resources are limited while online communication is crucial everywhere no matter when and how such recovery happens. However, ICT rescue supply tends to be for heavy duty, expensive, limited in availability, heterogeneous, and large in size (i.e., bigger than a suitcase). Our approach utilizes laptop computers and Linux Live USB flash drive that remain in a disaster affected site and can be easily carried by rescue workers. Our solution is plug-and-play to override the computing and communication resource of a laptop computer without accessing the existing storage drive on it. Our system provides ad-hoc networking, Wi-Fi hotspot, web server and various local online services for immediate use without requiring Internet connection.**

***Keywords–Post-disaster Networking; Linux Live USB; Wireless Mesh Network; Wi-Fi Hotspot; Local Online Services.***

## I. Introduction

One of the main challenges that victims and rescue workers face at disaster affected site is the lack of Information and Communication Technology (ICT) resources that makes organizing and exchanging information difficult. The consequence is that disaster mitigation and recovery might slow down and become inefficient. Victims and their relatives will get mentally stressed if they are not aware of the well-being of each other. Given the critical nature of this information, this paper proposes a self-contained software suite packed in Linux Live USB flash drive to enable ad-hoc ICT environment in a post-disaster situation.

The idea is to pack everything that helps local and, if available, global communication and information exchange in a disaster site into Linux Live USB. The proposed system has following features.

- *Light-weight and Small*: Our software suite is independent of any heavy and big machinery. A laptop computer is enough to boot from Live USB and start serving for other devices.
- *Inexpensive*: USB flash drives are inexpensive and easily available. Laptop computers need not have any special configuration. USB wireless cards are also available at low price.
- *Easy to setup*: Our system is preconfigured and plug-and-play. If any configuration is required on-the-fly, GUI helps users without special skills and knowledge.
- *Everything through Wireless*: Live USB node deploys Wi-Fi hotspot through which many user devices can communicate. Multiple Live USB nodes can establish wireless mesh network to form a larger network coverage. If Internet connection through LAN, 3G, satellite, etc. is available, then, user devices in the domain can also share the Internet connection through hotspot and mesh.
- *Local Online Services*: Our system provides local online services like web, proxy, video/audio communication, etc. within the local network. Once Live USB node is launched, users can start practical communication immediately without waiting for the recovery of Internet connection.

The contribution of this paper is compiling the widely demanded and accepted ICT technologies to be easily deployable, flexible, and available in a feasible manner for immediate use in a challenging situation like post-disaster recovery.

The rest of the paper is organized as follows. Section II presents some challenges and lessons learnt from ICT recovery activity in response to 2011 Tohuku Earthquake and Tsunami in Japan. Section III discusses the solution using Linux Live USB. Sections IV and V depict System Architecture and Implementation, respectively. Section VI deals with System Evaluation and Discussion. Section VII presents Related Work. We conclude the paper in Section VIII.

## II. Challenges in Post-Disaster Recovery of ICT Environment and Lessons from 2011 Tohoku Earthquake and Tsunami in Japan

We claim that the preparedness to collect, store and exchange information immediately after a disaster occurs is very crucial, whether the mode of communication is local network or Internet. As Utani et al. [1] mentioned, on-the-fly development of counter-disaster systems and crowdsourcing made great contributions. Based on their summary, we focus on the fact that deployment of such systems was very rapid and ready to accept inputs from disaster sites.

On the other hand, we also faced several challenges while post-disaster network recovery activity in 2011 Tohoku Earthquake and Tsunami in Japan [2]. Following is a summary of lessons learnt from the experience.

- An expensive system is not sustainable to support multiple distributed sites.

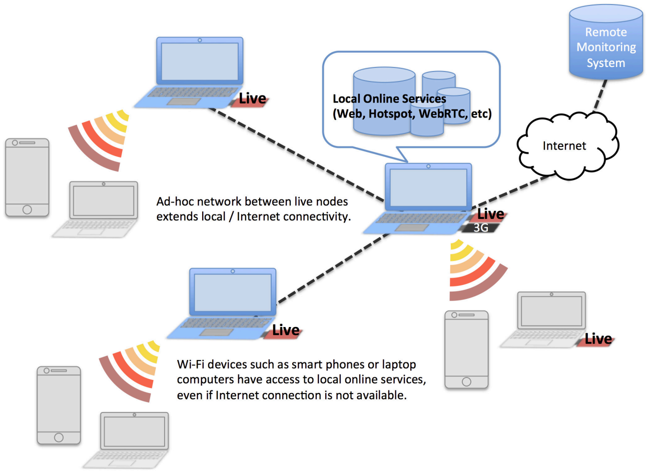

Remote Monitoring System
Local Online Services (Web, Hotspot, WebRTC, etc)
Internet
Live
Ad-hoc network between live nodes extends local / Internet connectivity.
Live
3G
Live
Live
Wi-Fi devices such as smart phones or laptop computers have access to local online services, even if Internet connection is not available.

Figure 1. Laptop computers booting from Live USB and connecting to each other forming mesh topology.

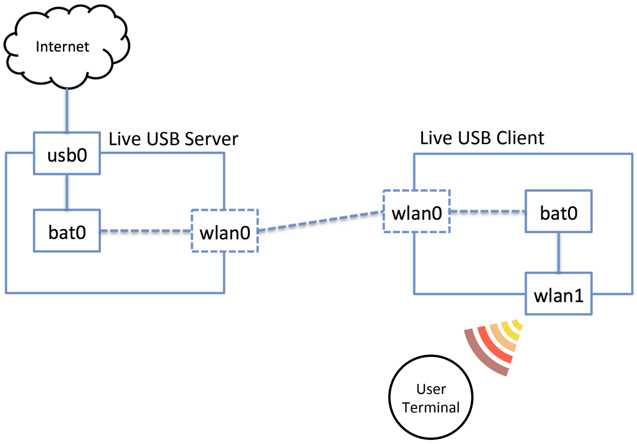


Figure 2. Internal connection of Live USB server, client and other devices.

- A big or heavy system introduces difficulty of logistics including packaging, dispatching, transportation, storage, planning and management.
- There might not be enough human resource to setup a technically complicated system. Again, handling multiple distributed sites will also be difficult.
- System should be up and running for a longer duration to reduce maintenance cost. Remote access is also important for health check and maintenance of the system.
- Some softwares/programs are written such that they need Internet connectivity to work even when local online communication is sufficient.
- It is not guaranteed that the Internet connection will be available soon.

We also made the following observations.

- Due to widespread use of smartphones, serving Wi-Fi hotspot helped people a lot.
- Thanks to the good preparedness for the disaster, power supply was sufficient to support additional devices. However, disaster sites faced power cuts.
- Laptop computers were available as popular rescue supply. However their maintenance cost is high. Who will conduct the update of operating systems and necessary software installation for each of them?
- Some desktop and laptop computers were there but not for public use. They were, of course, password protected and the data was untouchable. Hence, no one could use them.
- Mode of information dissemination is basically paper/notice board based in disaster sites. Digitizing such information and making it available online drastically improves the effectiveness of its dissemination.

## III. Solution using Live USB

Our system is packed in Live USB flash drive. Live USB allows users to boot guest Operating System (OS) without touching the existing one in a host computer. This means that our system makes use of the computing resource, and enables everything people want in a disaster site. As a USB flash drive is small and lightweight with enough storage capacity, it will solve most of the problems that we discussed in Sections I and II.

We raise three major benefits of using Live USB. First, the maintenance of software suite such as installing, deleting, upgrading and testing applications and the guest operating system can be done at a single place. So, users can save a huge amount of time and network bandwidth for setting up the disaster rescue mode of Operating Systems. Second, users need not worry about what Operating System is installed on the laptop computer. They need not have IT skills to handle available features as User Interface can be designed to be beginner-friendly. There is no need to know the password, if any, of the host Operating System installed on laptop computer. Almost any laptop computer can be used unless that system's Basic Input/Output System (BIOS) is password protected. Third, as the behavior of Operating System can be controlled, a Live USB node can be configured to connect to remote system, also knows as Virtual Private Network (VPN) server, for remote login. This VPN server is present in global Internet, and Live USB node connects to it if and only when it can access Internet. VPN connection will enable human resources such as IT engineers to stay outside of disaster sites without needing to travel too much. If the remote login does not help in trouble shooting, the simple solutions are system reboot or replacement of the USB flash drive, that are not beyond IT skills of people in a disaster site.

### A. *Expected Deployment Scenario*

Figure 1 presents the expected deployment scenario of our system. Laptop computers boot using Live USB flash drives that are carried to or prepared in a disaster affected site. The guest OS is pre-configured to support wireless mesh networking. Once the guest OS is booted, Wi-Fi devices of laptops try to connect to neighbor Live USB nodes. On some of the laptops, USB wireless cards can be used to create Wi-Fi hotspot, so that smartphones, laptops, sensors and any other Wi-Fi devices can also connect to the network. As an initial expectation, Live USB nodes will not move around. They are most probably deployed in a public space like evacuation

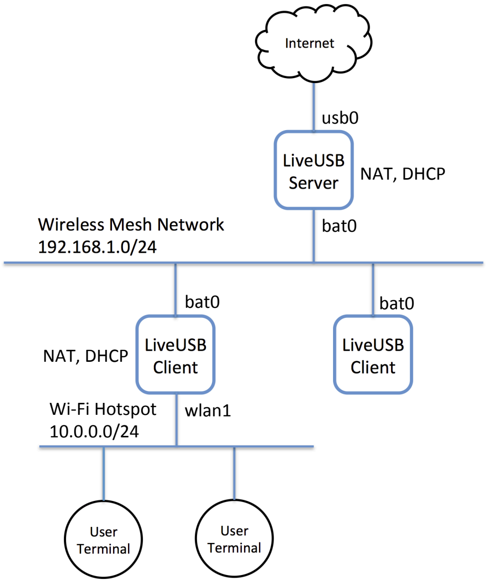


Figure 3. Network Diagram of Live USBs and other devices.

buildings and shared by users in turn, while providing various online services including Wi-Fi hotspot.

## IV. SYSTEM ARCHITECTURE

### A. *Live USB Server and Client*

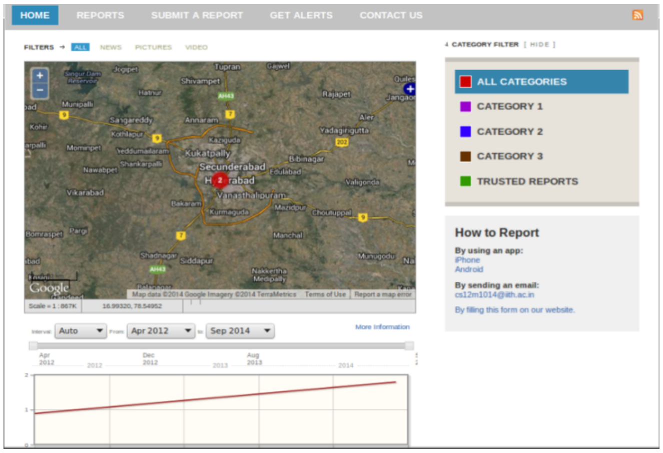
A Live USB node acts either as a server or a client, depending on what packages it contains. The server provides local online services such as web, online storage, video/audio/text communication, etc. Only one Live USB server is active at any time in one disaster site. Live USB server can have Internet connectivity if it is available at disaster site. The server also creates a wireless mesh network to let other Live USB clients (or simply client) access the local online services and share the Internet connection. Thus, server acts as the gateway to Internet. Live USB server (or simply server) connects to VPN server so that it can be monitored remotely.

A Live USB client avails the functionalities provided by Live USB server. Depending on the needs at a disaster site, client connects to the server through wireless mesh network and accesses the local online services. It can also share the Internet connection, if it is available to Live USB server.

### B. *Network Design*

Figure 2 depicts the internal connection of Live USB server, client and other devices. Our system works without an existing wireless or wired LAN infrastructure. Live USB nodes enable wireless mesh network to communicate with each other and share the local and Internet connection. As wireless mesh network forms a single layer 2 domain, it is important to reduce the amount of unwanted traffic.

All the Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) requests from Live USB clients are addressed by Live USB server. Server also acts as the Network Address Translation (NAT) router for the clients. On the other hand, Live USB client addresses DNS and DHCP requests from user terminals that are connected to it using Wi-Fi hotspot. Live USB server can also serve Wi-Fi hotspot, if required. bat0 (Figure 3) interface of Live USB server has static IP address so that all the clients and user terminals can access services provided by the server.



Interval: Auto From: Apr 2012 to: Sep 2014

How to Report

By using an app:
iPhone
Android

By sending an email:
cs12m1014@iith.ac.in

By filling this form on our website.

Figure 4. Location Based Report Submission in Ushahidi.

### C. *Immediate Deployment of Local Online Services*

To immediately enable ad-hoc communication and information exchange among users, Live USB server provides local online services that are pre-configured and ready to be deployed in a disaster affected site. Instead of installing variety of applications on each Live USB individually, our system integrates most of the applications on Live USB server that can be accessed by clients and other devices. The following two features introduce a lot of benefits that make ICT service and environment to be prepared for instantaneous information exchange.

- It reduces the maintenance cost of individual software client.
- By locally hosting services at Live USB server, these applications are "Internet-free" or offline at the moment of deployment. It will also save bandwidth.

Once the Internet connection becomes available, Live USB nodes and other user terminals can switch to "Internet-dependent" which is online mode.

## V. IMPLEMENTATION

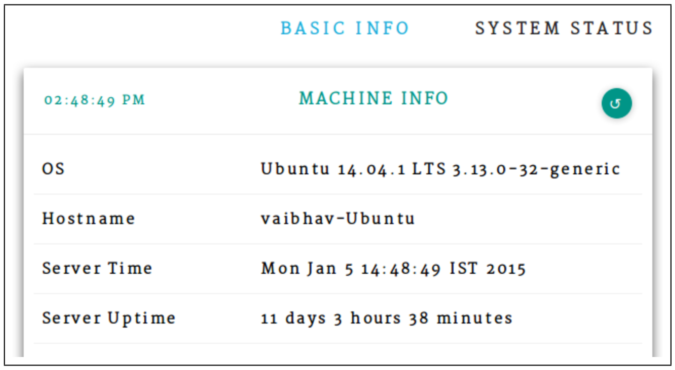
Server provides various services that are not required to be present on client. In order to reduce the size of client OS image, we prepare separate guest OS images for implementing server and client mode of Live USB. Table I summarizes the implementation details of the system.

### A. *Web Based Information Exchange*

Web-based content management systems like MediaWiki [7], WordPress [8] and Ushahidi [9] (Figure 4) help to facilitate local information exchange. All these systems can be accessed via web browser and require single click operation. Ushahidi works only in online mode as it requires the exact GPS location of the sender of the report. Currently, work is being done to make it work in offline mode as well.

TABLE I. IMPLEMENTATION DETAILS OF LIVE USB SERVER AND CLIENT

| **Feature** | **Live USB** | |
|---|---|---|
| *Hardware* | SanDisk Extreme USB 3.0 32 GB USB Flash Drive | |
| *Guest OS* | Linux Ubuntu 14.04.1 LTS | |
| *Mesh Networking* | Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N.) Advanced [3] | |
| *Wi-Fi Hotspot* | hostapd [4] and dnsmasq [5] | |
| | **Live USB Client** | **Live USB Server** |
| *Persistent Storage* | 4 GB | 8 GB |
| *Web Proxy* | - | Squid [6] |
| *Monitoring and Health Check* | Live USB clients that are connected to the server can be monitored at the server. | Live USB servers can be monitored at Remote VPN server when server can access Internet. |


BASIC INFO SYSTEM STATUS

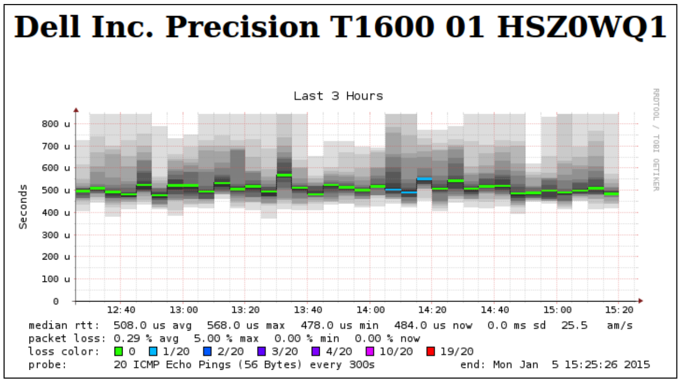
02:48:49 PM MACHINE INFO

| | |
|---|---|
| OS | Ubuntu 14.04.1 LTS 3.13.0-32-generic |
| Hostname | vaibhav-Ubuntu |
| Server Time | Mon Jan 5 14:48:49 IST 2015 |
| Server Uptime | 11 days 3 hours 38 minutes |

Figure 5. Linux Dash showing various statistics of Live USB Servers.

### B. Real-Time Multimedia Communication

Our system enables audio, video and text chat using WebRTC [10]. WebRTC works by enabling Peer-to-Peer (P2P) connection between clients. In case, P2P connection is not successful (mainly due to NATs and/or firewalls), it falls back to Server-Client model by relaying data through some server in global Internet [11]. Ongoing work includes installing STUN server on Live USB server so that basic text, audio and video communication can take place without requiring Internet connection.



```
median rtt:   508.0 us avg   568.0 us max   478.0 us min   484.0 us now   0.0 ms sd   25.5   am/s
packet loss:  0.29 % avg   5.00 % max   0.00 % min   0.00 % now
loss color:   🟩 0   🟦 1/20   🟦 2/20   🟪 3/20   🟪 4/20   🟪 10/20   🟥 19/20
probe:        20 ICMP Echo Pings (56 Bytes) every 300s                 end: Mon Jan  5 15:25:26 2015
```

Figure 6. Once Live USB server successfully connects to VPN server, its entry gets automatically added as a monitoring target of SmokePing.

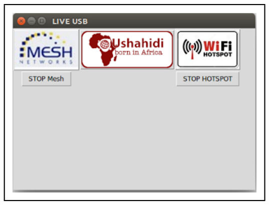
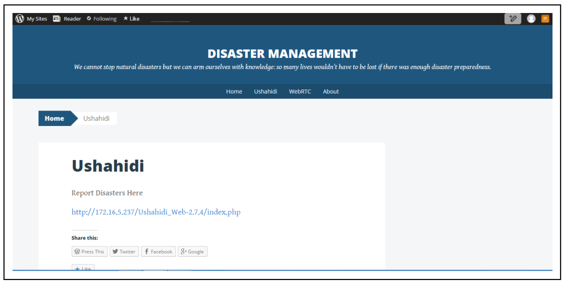


Figure 7. Server LiveUSB GUI for starting/stopping various services like Mesh Networking, Ushahidi and Hotspot.



DISASTER MANAGEMENT

*We cannot stop natural disasters but we can arm ourselves with knowledge: so many lives wouldn't have to be lost if there was enough disaster preparedness.*

Home Ushahidi WebRTC About

Home Ushahidi

Ushahidi

Report Disasters Here

http://172.16.5.237/Ushahidi_Web-2.7.4/index.php

Share this:

Figure 8. Client LiveUSB WordPress interface for WebRTC and Ushahidi.

### C. System Monitor and Health Check

Each Live USB node keeps track of latest information such as memory, CPU, disk, I/O, swap, logged in accounts, common applications, ping speed etc. Linux Dash [12] is a simple web based dashboard that provides these statistics (Figure 5). Live USB server keeps track of all the Live USB clients that are connected to it. It provides a link to each of the client so that each client's latest statistics can be monitored individually. On the other hand, Live USB server keeps attempting to connect to Remote VPN server, until it is successful in doing so. The script at the VPN server automatically adds an entry for that particular Live USB server in SmokePing [13] (Figure 6). VPN server keeps track of all the Live USB servers that are connected to it. Thus, Live USB servers can be monitored individually by VPN server. We send only minimal data to the VPN server to avoid unnecessary consumption of Internet bandwidth. Some engineering was done to manage the battery life of Live USB server node. When server's battery goes critically low, it sends a message to remote VPN server and goes in hibernation mode. VPN server then sends intimation to the email id and/or contact no. provided by user. User can then charge the battery of laptop to which Live USB server is connected or use some other laptop computer as server.

### D. GUI for Live USB Management

System GUI (Figure 7) contains some controllable features which help in setting up of the Live USB services like Mesh Network, Ushahidi and WordPress (Figure 8).

## VI. Evaluation and Discussion

### A. Tested Hardware

We have confirmed that hardware summarized in Table II supports wireless mesh networking and Wi-Fi hotspot configured on the guest OS. All the testing was done using 5.18GHz frequency band (channel 36).

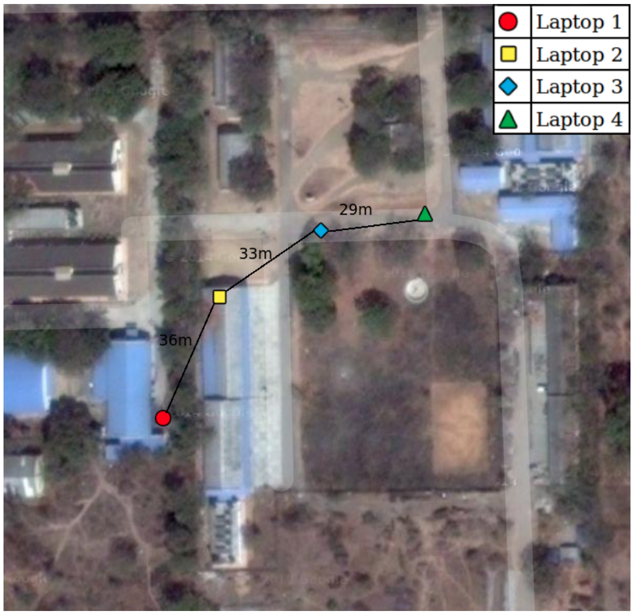


Figure 9. Multi-hop Scenario in IIT Hyderabad Hostel Premises.

TABLE II. VERIFIED HARDWARE AND DRIVERS

| **Laptop Model** | **Internal Wi-Fi Model** |
|---|---|
| Acer Aspire 5750G | Atheros AR5B97 802.11b/g/n |
| Sony VAIO E Series | Atheros AR9285 VPCEB14EN 802.11b/g/n |
| HP Pavilion G6 1200tx | Broadcom 4313GN 802.11b/g/n |
| **Wireless LAN Adapter Model** | **Drivers Used** |
| Logitec LAN-W150N/U2 | rt2800usb |
| Buffalo WLI-UC-AG300N | rt2800usb |

### B. One-hop Scenario

In one-hop scenario, two laptop computers were kept at a distance of around 50m from each other. It was made sure that they both were in line-of-sight of each other and there was no interference with neighbor devices. The test was performed in an open field in IIT Hyderabad campus. TCP throughput in this case was observed to be 17.7 Mbps and UDP throughput was 28 Mbps. As there is no interference in this scenario, these throughput values act as the benchmark for multi-hop scenario.

### C. Multi-hop Scenario

In order to verify the performance of mesh in multi-hop scenario, we created a mesh of 4 laptop computers. The transmission power of Wi-Fi NIC was manually managed to make sure that only adjacent laptops (Figure 9) are in direct line-of-sight of each other. Since this test was conducted in hostel premises, there was interference as a lot of people were moving in between laptop computers. Iperf [14] server was enabled on Laptop 1 and Iperf client was used to test the throughput from each of the subsequent laptop to Laptop 1. TCP throughput was averaged over 3 runs, while in case of UDP, we consider the minimum, maximum and average throughput by changing the rate of data transfer. Figure 10 presents the results of the setup.

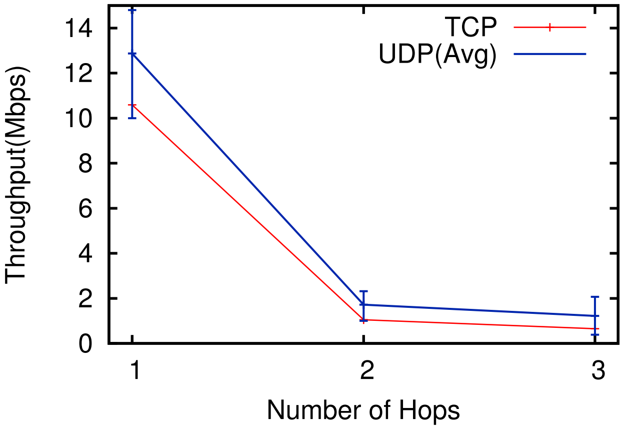


Figure 10. Results of Multi-hop Scenario.

### D. What is a good SSID for Wi-Fi hotspot?

In order to well deliver our network connectivity and services to users, currently, we are using the unified SSID “00000INDIA”. As in most cases SSIDs are sorted in alphabetical order, five zeros will contribute to place our SSID at the top of list. The choice of this SSID is inspired by discussions and guidelines across telecom networks [15] after 2011 Tohoku Earthquake and Tsunami in Japan. The discussion was about how to enable free and public Wi-Fi hotspots in disaster situation where a lot of commercial ISPs may provide their own hotspot services under different configurations. The discussion also covered the responsibility and other operational guidelines for operating public Wi-Fi hotspots.

### E. Does our system replace existing solutions?

Our system does not aim to challenge existing post-disaster ICT environment and technologies. But it provides an alternative when the existing ones are not available. Some components of Live USB, such as local online services, might be easily integrated with the other systems as add-on.

### F. Security in Open Network

Our network is basically open for everyone. However, such a network often faces security issues. In our deployment scenario, Live USB server and client should have mechanisms for security inspection and access control before letting user terminals start communication through Wi-Fi hotspot.

### G. How do we address the wireless quality problem?

Our system is always subject to the quality of wireless communication as it works by creating single large mesh network. Some optimization can be done to enable multi-channel support using multiple USB Wi-Fi NICs on each Live USB node. This approach will require some engineering so that both the Wi-Fi NICs on same Live USB node are not configured in same channel. Also, giving cognitive feature to Live USB nodes will ease the performance degradation if a lot of Wi-Fi hotspots are deployed in a single place [16].

## VII. Related Work

Mobile ad-hoc networking and wireless mesh networks have been actively proposed and examined for post-disaster recovery. [17][18][19][20] discussed communication network using MANET, where highly demanded protocols and applications, like flooding-based communication protocol, push to talk and telemedicine, are made available in post disaster situation. [21] examined Multicast in MANET to efficiently disseminate information. [22] introduced Wireless Mesh Network that can be used in both daily life and emergency situations. [23] proposed to make use of survival time of wireless sensor networks and evacuate critical data to safe zone. [24] examined combination of Overlay Network and MANET to provide redundancy and continuity of services. Their solution expects the availability of high speed network to appropriately place mirror services through overlay network. [25] puts forward a session-based mobility management in MANET which meets the requirements of rapid deployment of the network with auto-configuration.

There has been significant amount of work done on recovering the ICT environment at a disaster site. [26] presented a fully-functional prototype of long-distance multimedia wireless mesh network during the time of large-scale disaster, but their solution is aimed at networking in open space, where mesh nodes may move around. [27] discussed the use of cognitive agents for bringing back telecommunication network in post-disaster aftermath. The cognitive agents run periodically to detect emergency situation, if exists. In case of emergency, the agents try to disseminate information to disaster information center. [28] proposed a quickly-deployable package for post-disaster communication, and [2] reported how such systems were used in 2011 Tohoku Earthquake and Tsunami in Japan. Their approach requires technically trained people for the setup.

## VIII. Conclusion and Future Work

This paper considered the challenges of having easy access to computers and communication infrastructures in post-disaster settings. It proposes a solution using Linux Live USB flash drive, where the guest OS and a set of preloaded software are ready to serve. One can attach the USB flash drive to an available computer, and boot her own OS to get access to the computing resources. A Live USB node can also act as a Wi-Fi hotspot in a wireless mesh setting. Our system is intended to fill the gap between the timings when people in a disaster site start to demand the ICT infrastructure and when such infrastructure is actually recovered and becomes ready to use. The result of field trial proved that our system can facilitate the online services and enable user terminals to access them through Wi-Fi hotspot and mesh network with or without the Internet connection.

So far, we have tested a very simple deployment scenario. The traffic engineering and optimization of backhaul connection under complex network topology, like supporting multiple Internet connections, will be future work to directly improve the network capacity. Also, some simulations can be conducted to explore the appropriate size of network for larger-scale deployment. As "crowdsourcing" provides a scaling out solution for data processing, data can be exported from the disaster site and made available for third-party software developers or data analysts. As part of software management of Live USB, we need to examine the mode of information exchange which may vary by emergence of new technology or change of trend.

## References

[1] A. Utani, T. Mizumoto, and T. Okumura, "How geeks responded to a catastrophic disaster of a high-tech country: Rapid development of counter-disaster systems for the great east japan earthquake of march 2011," in Proceedings of the Special Workshop on Internet and Disasters, ser. SWID '11. New York, NY, USA: ACM, 2011, pp. 9:1–9:8. [Online]. Available: http://doi.acm.org/10.1145/2079360.2079369

[2] K. Kataoka, K. Uehara, O. Masafumi, and J. Murai, "Design and deployment of post-disaster recovery internet in 2011 tohoku earthquake," IEICE transactions on communications, vol. 95, no. 7, 2012, pp. 2200–2209.

[3] Open-mesh, "B.A.T.M.A.N. Advanced Documentation Overview," http://www.open-mesh.org/projects/batman-adv/wiki/Doc-overview, [Accessed: 2015-03-19].

[4] Linux Wireless, "Hostapd," http://wireless.kernel.org/en/users/Documentation/hostapd, [Accessed: 2015-03-19].

[5] "Dnsmasq," http://www.thekelleys.org.uk/dnsmasq/doc.html, [Accessed: 2015-03-19].

[6] "Squid: Optimising Web Delivery," http://www.squid-cache.org/, [Accessed: 2015-03-19].

[7] "MediaWiki," https://www.mediawiki.org/wiki/MediaWiki, [Accessed: 2015-03-19].

[8] "WordPress," https://wordpress.org/, [Accessed: 2015-03-19].

[9] "Ushahidi," http://www.ushahidi.com/, [Accessed: 2015-03-19].

[10] "WebRTC," http://www.webrtc.org, [Accessed: 2015-03-19].

[11] Mozilla Corporation, "WebRTC Connectivity," https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Architecture/Connectivity, [Accessed: 2015-03-19].

[12] "Linux Dash," http://linuxdash.afaqtariq.com/, [Accessed: 2015-03-19].

[13] "SmokePing," http://oss.oetiker.ch/smokeping/, [Accessed: 2015-03-19].

[14] "Iperf," https://iperf.fr/, [Accessed: 2015-03-19].

[15] "Guideline About Large-scale Deployment of Free Open Public Radio LAN In Disasters Occurrence," http://www.wlan-business.org/info/pdf/Wi-Fi_Free_Guideline_v1.01_20140527.pdf, [Accessed: 2015-03-19].

[16] B. Tamma, B. S. Manoj, and R. Rao, "An autonomous cognitive access point for wi-fi hotspots," in Proceedings of the Global Telecommunications Conference (GLOBECOM), Nov 2009, pp. 1–6.

[17] T. Umedu, H. Urabe, J. Tsukamoto, K. Sato, and T. Higashinoz, "A manet protocol for information gathering from disaster victims," in Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom), March 2006.

[18] Y.-N. Lien, H.-C. Jang, and T.-C. Tsai, "A manet based emergency communication and information system for catastrophic natural disasters," in Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS), June 2009, pp. 412–417.

[19] Y.-N. Lien, L.-C. Chi, and C.-C. Huang, "A multi-hop walkie-talkie-like emergency communication system for catastrophic natural disasters," in Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW), Sept 2010, pp. 527–532.

[20] J. K. et al., "Development of mobile ad hoc network for emergency telemedicine service in disaster areas," in Proceedings of the International Conference on New Trends in Information and Service Science (NISS), June 2009, pp. 1291–1296.

[21] M. Iqbal, X. Wang, D. Wertheim, and X. Zhou, "Swanmesh: a multicast enabled dual-radio wireless mesh network for emergency and disaster recovery services," Journal of Communications, vol. 4, no. 5, 209, pp. 298–306.

[22] Y. Takahashi, Y. Owada, H. Okada, and K. Mase, "A wireless mesh network testbed in rural mountain areas," in Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization. ACM, 2007, pp. 91–92.

[23] C. Debata and R. Roy, “Efficacy of power of two choices technique for data evacuation process in sensor networks for post-disaster relief operations,” in Proceedings of the IEEE International Conference on Signal Processing, Computing and Control (ISPCC), Sept 2013, pp. 1–6.

[24] Y. Shibata, H. Yuze, T. Hoshikawa, K. Takahata, and N. Sawano, “Large scale distributed disaster information system based on manet and overlay network,” in Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW), June 2007, pp. 33–33.

[25] R. Li, R.-L. Yang, and D.-W. Hu, “A designing of mobility management mechanism in manet in disaster-rescue situations,” in Proceedings of the 11th IEEE International Conference on Communication Technology (ICCT), Nov 2008, pp. 596–599.

[26] K. Kanchanasut, A. Tunpan, M. Awal, T. Wongsaardsakul, D. Das, and Y. Tsuchimoto, “Building a long-distance multimedia wireless mesh network for collaborative disaster emergency responses,” Internet Education and Research Laboratory, Asian Institute of Technology, Thailand, 2007.

[27] S. Majid and K. Ahmed, “Post-disaster communications: A cognitive agent approach,” in Proceedings of the Seventh International Conference on Networking (ICN), April 2008, pp. 645–650.

[28] K. Kataoka, K. Uehara, and J. Murai, “Lifeline station: A quickly deployable package for post disaster communications,” in Proceedings of the Internet Conference, 2009, pp. 41–47.

# The Percolation Theory Based Analysis of Data Transmission Reliability via Data Communication Networks with Random Structure and Kinetics of Nodes Blocking by Viruses

Dmitry Zhukov
Institute of Information Technologies
Moscow State Technical University, MIREA
Moscow, Russia
e-mail: zhukovdm@yandex.ru

Sergey Lesko
Institute of Information Technologies
Moscow State Technical University, MIREA
Moscow, Russia
e-mail: sergey@testor.ru

*Abstract*—**In the paper, open global computer networks and data communication networks are considered as structures with a random topology. Processes of epidemics spreads are described by percolation theory. "The percolation thresholds", fraction of blocked nodes at which the whole network loses its working capacity, are calculated for different numbers of communications per node. For the real data communication networks with the average number of communications per node in the range of 2.5 to 3.5, the share of the used equitype equipment and the software types should not exceed the margin from 48% to 63%.**

***Keywords: data communication network; blocking nodes; network topology, the percolation threshold; virus distribution dynamics***

## I. INTRODUCTION

An important task in ensuring reliable functioning of data communication networks, as well as protection of the transferred information, is the study of the formation of groups of network node physically connected by communication channels but blocked (excluded from operation) for some reason. For example, blockage is possible during computer viruses epidemics. Under certain conditions, such groups of blocked nodes can increase in size and form clusters, which can lead to an overall loss of functionality of the data communication network. For instance, a cluster can form when there is some blockage of a backbone node of data network at the regional or city level. Alternatively, a cluster can originate in a base station of a mobile network as a result of peak load or overload, or when there is a computer virus epidemic in computer networks, which blocks the operation of different network equipment. Our objective is to develop a model describing the processes of nodes clustering based on the percolation theory, the main assumptions of which will be stated further on.

Historically, any data communication network starting from the city region level has an irregular random structure. The brightest example of such a network is the Internet. This is caused by many factors among which we can single out the following: providers having different network and communication equipment, a fluctuating number of subscribers with constantly changing connection topology and many others.

At present time, spreading of epidemics is often described as a process with structure similar a Kailey tree with random number of connections [1]-[2].

One can pay closer attention to a number of works by R. Pastor – Satorras and A. Vespignani, where the authors study the problem of defining the probability of infection depending on the node distance from the source of threat in networks of different scale and with varying number of nodes [3]–[7]. The authors used the scale and number of nodes as topological parameters; however, there were no special insights into the diversity of networking structures and the blocked nodes clustering.

In common case a scale free graph can have any number of nodes. Figure 1a shows such a graph with the total number of nodes equaling 100.

The description of virus epidemic topology using a scale free graph model produces interesting results. However, at some stage, infected network nodes can start sending copies of viruses to already infected nodes, and the process topology will look as shown in Figures 1b and 1c [1].With the help of a scale free graph model, we can consider the data transfer traffic dynamics [8], [9], as well as the processes of network structure hierarchical growth [10].

Obviously, if the amount of blocked nodes is not too large, there will be an "open" route (a way formed by unblocked nodes) between two randomly selected nodes located at a distance. We will refer to the amount of blocked nodes at which the network becomes nonfunctional as a *percolation threshold*–the network will be functional below this value despite the fact that it contains some nodes or their groups (clusters) blocked by viruses. Above the percolation threshold, the whole network turns off and loses its data transfer functionality. There is no "open" way between two randomly selected nodes.

Studying processes of blocked nodes clusters formation and data percolation in networks with different (including random) topology has a lot of scientific and practical importance for the development of topology of data communication networks having high fault-tolerant features.

It would greatly help improve their technical and economic, as well as operational characteristics, and create

ISBN: 978-1-61208-404-6

new methods and methodologies of computer network and applications protection.

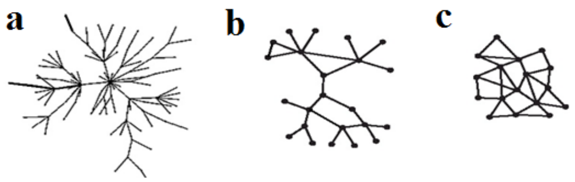


Figure 1. Different types of a scale free graph [2]: (a) – general case, (b)- the beginning of mutual DDoS attacks,(c) – the process of mutual DDoS attacks becomes considerable.

Besides, we should distinguish between two notions:

- Physical connections between the nodes. Two nodes are considered neighbors if they have a direct (without an intermediary) communication channel.
- Address linkage between the nodes. A virus can send its copy to a randomly selected node with a random IP-address instead of sending it to its physical neighbor.

In the latter case, the virus epidemic development topology looks like the Kailey tree (network) with a random number of communications, while in the former, the structure of physically connected infected nodes will be more complex and it has almost never been studied.

In Section II, we prove a choice of object and research methodology.

In Section III, we provide the description and discussion of results for data transmission modeling received in framework percolation theory for networks with random topology.

In Section IV we state the main conclusions drawn on the basis of the results received in the work.

## II. Subject and Study Methodology

We base our choice of network structure for a complex study of topology influence on its reliability on the fact that assessment of *real networks similarity* and different theoretical types of topologies on the basis of modeling can help single out a network (or types of networks) with the features closest to those of real networks (for example, the Internet), which is important for analysis of processes happening in the existing networks and ensuring their reliability.

Figure 2 shows the map of a mobile network operating in one of the Russian Federation regions.

The given map shows that real networks of data communication have a random structure similar to the one shown in Figure 1c; therefore, this article scrutinizes the random network with a set of communications per node.

As mathematical apparatus of the conducted research, we used the percolation theory, its basics being represented in [11]-[15].

During the modeling, we made the following assumptions: all nodes ($10^6$) of a computer network create a single network with a specific topology. Blocking of nodes occurs when infected with a computer virus. The virus can send its copies ($10^2$) from any node to any other arbitrary node (with probability of infection of $5\cdot10^{-3}$) by selecting its address from the entire set of address space (not necessarily physically connected nearby sites). At the next steps of the epidemics, the infected nodes are sending copies of the virus to other nodes in the network etc.

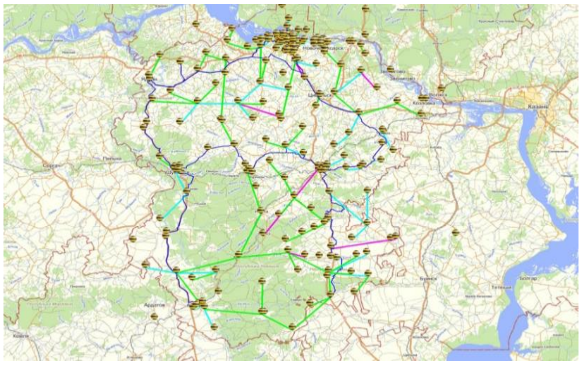


Figure 2. The location of base stations in the north of Chuvashia round the capital – the city of Cheboksary.

The average cluster size of blocked sites was determined at each step by numerical modeling methods; the infection process was carried out until the network reached the percolation threshold.

From a mathematician’s point of view, the percolation theory should be attributed to the probability theory in graphs. The most widespread problems of the percolation theory are the *lattice problems*, viz. the node problem and the connection problem. Let us consider a continued square grid. We shall name the points of line crossing *nodes (vertexes* of the graph); the lines themselves will bear the name of *communications (graph edges)*.

In the connection problem one tries to find an answer to the following: which share of communications should be eliminated (cut off) for the net to fall into two equal parts? In the node problem the nodes are blocked (removed, all the connections with the node being cut off) and one searches for the share of blocked nodes leading to network falling apart. In the percolation theory, a chain of connected items is called a *cluster*. A cluster connecting two opposite sides of the system is dubbed *percolating*, *infinite*, *spanning* or *connecting*. Below the percolation threshold, there are only clusters of a finite size.

The staff members of IBM R&D Centre (Scott Kirkpatrick, Winfried Wilcke, Robert Garner, and Harald Huels) studied the possibility of applying the percolation theory to the data storage systems [16]. They proposed the following model. A data cube of 1000 base elements connected by a cubic-cell type contained two types of cells (nodes), viz. the ones containing the immediate data and the cells (nodes) ensuring fault-tolerance – data replication. Since each node of such a system should not only provide the data output but also ensure data passage through a storage array (the access to other data), it was reasonable to employ the percolation theory. The use of the percolation theory allowed proving that it is enough to have just one copy of replicated data to ensure continuous fault-tolerant operation of the network. In case of excessive replication (two or more copies of data), one could observe a trespass over the percolation threshold, formation of non-conducting cluster in a cubic lattice, which led to the system operation failure. This model was put to good use in the system of data storage 'Ice cube' supplied by IBM Company and allowing the creation of a 32-terabyte array of data.

There are no analytical models elaborated to describe the percolation processes and to study random networks with multiple communications. Their research is possible only by numerical methods of modeling. For this purpose, at first it is necessary to construct a structural model of a network (see Figure 3), then, to choose a couple of any arbitrary nodes and using numerical modeling methods to define at what part of unblocked nodes in the considered network there is a freeway between A and B nodes. Then, this procedure is likewise performed for any other couples of nodes (in our case for the couples of C and D, E and F nodes in Figure 3 etc.). After that, with statistical averaging the results of separate experiments, we determine the average value of percolation threshold for all considered couples of nodes.

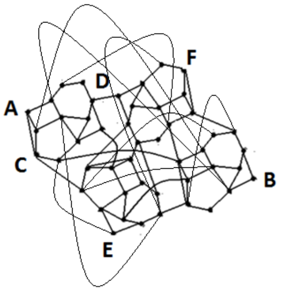


Figure 3. Data transmission random network structure.

## III. RESULTS AND DISCUSION

### A. *Percolation (flow) of information in a random network of data communication*

Table 1 presents the results of numerical modeling to find percolation threshold for random networks with the set of ways between nodes (see Figure 3) and various averages of communications per node.

With an increasing average number of communications per network node, the time and computing resources consumption significantly increases as well. For this reason, we had to choose the number of communications per node ranging from 2.5 to 15 in our numerical modeling.

In Figure 4 the dependence of the results given in Table 1 is shown. The percolation threshold decreases monotonically to 0.115 with the growth of the communications average per one network node. There is really no need to carry out numerical modeling at great values of average of communications per node, and it is possible to extrapolate the results onto the area of great values.

The graphical type of dependence in Figure 4 is similar to exponential law, therefore it can be described by function: $P(x) = P_0 e^{-z}$, where P(x) is the percolation threshold value at the average of communications per node equaling some value x, z=1/x; $P_0$ is the percolation threshold valueat an infinitely large number of communications per node.

TABLE I. PERCOLATION THRESHOLDS FOR RANDOM NETWORKS

| **Network type** | **Average number of communications per one network node** | **Fraction of the activated nodes at which there is conductivity in the network ($n_c$ - percolation threshold)** |
|---|---|---|
| ***A random network with a set of paths between nodes*** | 2.36 | 0.515 |
| | 2.82 | 0.425 |
| | 3.29 | 0.365 |
| | 4.70 | 0.270 |
| | 4.75 | 0.250 |
| | 6.15 | 0.150 |
| | 6.17 | 0.185 |
| | 6.75 | 0.175 |
| | 9.41 | 0.170 |
| | 10.02 | 0.150 |
| | 10.31 | 0.130 |
| | 10.69 | 0.135 |
| | 11.07 | 0.115 |
| | 13.10 | 0.115 |

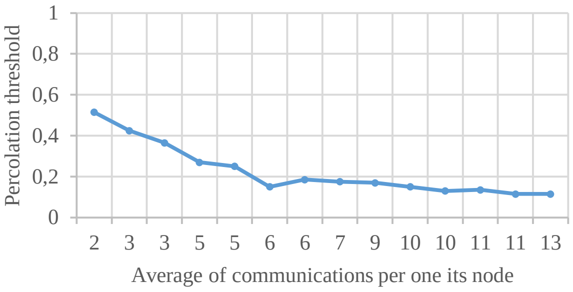


Figure 4. Dependence of percolation threshold size in random network on the average of communications per its one node.

As Figure 5 reveals, the data presented in Table 1 are well linearized in coordinates: lnP(x) depending on z=1/x (a natural logarithm of the percolation threshold is an inverse value to the average of communications x per node) that confirms the possibility of using the function: $P(x) = P_0 e^{-z}$.

Points in Figure 5 mark the experimental data, and the solid line corresponds to the linear dependence:
y = 4.39z-2.41, with a big value of correlation coefficient that equals 0.95.

At z=1/x=0 (corresponds to the case x = ∞) we receive: y=ln$P_0$ =-2.41, and the value of the percolation threshold at infinitely large number of communications per $P_0$ node will be equal 0.09.

It should be noted that, logically, it has to tend to 0; however, the obtained result can be explained as follows. At a very large number of communications, there can be a change in the law of dependence of percolation threshold on the number of communications. Nevertheless, it is possible to claim that the received dependence remains fair for random networks with significantly large number of communications per node. Thus, for a random network with an infinitely large number of communications per node it is enough to have the fraction of the activated nodes equal to 0.09 from the total number so that there could appear a carrying-out chain of nodes and the network could solve the set information task. At the average of communications equaling 100, the threshold of a percolation will amount to 0.094, and at 10 to 0.139 respectively.

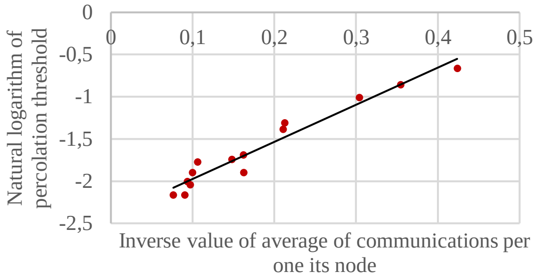


Figure 5. The logarithmic dependence of percolation threshold value in a random network on the inverse value of average of communications per its one node.

In practice, the number of communications of a random network per node will from 2.5 to 3.5, which yields the percolation threshold values ranging from 0.52 to 0.37. We calculated it using the data given in Figure 5 and the equation = 4.39z-2.41

## B. *Clustering of a random network*

If we consider the transition of any node of a random network from the efficient (not infected) state in the blocked state as a random process (with some probability of transition), this probability has to influence the average size of a cluster (a group of the nodes directly interconnected) of the blocked nodes.

By numerical experiments, we studied the influence of blocking probability on the average size of blocked nodes cluster of random networks with various average numbers of communications per node. Network with 10000 nodes were investigated. Two limit cases were considered: one for a small average of communications per node of a random network (see Table 2 and Figure 6) and the other limit case is for a big average of communications.

In Figure 6, curve 1 corresponds to the average of communications per node of a random network equaling 2.13; curve 2 depicts *idem* equaling 2.53; curve 3 is for 2.80 and curve 4 outlines for value 3.27 respectively. As the data in Table 2 and in Figure 6 demonstrate, the size of the cluster of the blocked nodes depends on the average of communications and probability of infection. With the growth of the average of communications, at the fixed probability of blocking, the size of the cluster increases. We can observe a similar situation with a great average of communications per node of a random network.

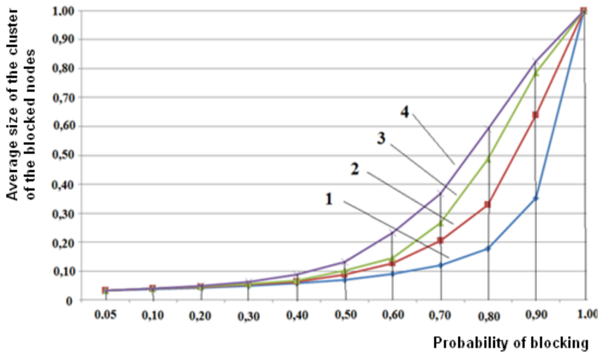


Figure 6. Dependence of the average size of the cluster of the blocked nodes on probability of blocking (for example, infections with a virus).

TABLE II. DEPENDENCE OF THE AVERAGE SIZE OF THE CLUSTER OF THE BLOCKED NODES ON PROBABILITY OF BLOCKING

| **Probability of blocking** | **The average size of the blocked nodes cluster (in fractions relative to the total number)** | | | |
|---|---|---|---|---|
| | ***For the network with an average of communications per node equaling 2.13*** | ***For the network with an average of communications per node equaling 2.53*** | ***For the network with an average of communications per node equaling 2.80*** | ***For the network with an average of communications per node equaling 3.27*** |
| 0.05 | 0.033 | 0.033 | 0.033 | 0.033 |
| 0.1 | 0.037 | 0.039 | 0.039 | 0.039 |
| 0.2 | 0.041 | 0.044 | 0.047 | 0.049 |
| 0.3 | 0.049 | 0.054 | 0.056 | 0.062 |
| 0.4 | 0.059 | 0.062 | 0.067 | 0.087 |
| 0.5 | 0.070 | 0.087 | 0.101 | 0.132 |
| 0.6 | 0.090 | 0.126 | 0.146 | 0.233 |
| 0.7 | 0.120 | 0.206 | 0.268 | 0.368 |
| 0.8 | 0.178 | 0.330 | 0.487 | 0.591 |
| 0.9 | 0.353 | 0.638 | 0.785 | 0.825 |
| 1 | 1 | 1 | 1 | 1 |

## C. *Kinetics of blocking nodes in the address space of computer networks and achieving the percolation threshold*

Currently, researchers recourse to empirical susceptible–infectious (SI) and susceptible–infectious–recovered (SIR) models originating from biology [17] to describe the kinetics of infection of data communication networks. However, instead of the empirical ones, some more reasonable mathematical and information models are required for the adequate description of the blocking processes.

To create such a model, we have considered the network consisting of L computers, for example, in which viruses can reproduce with coefficient of reproduction equal to ξ. Viruses start spreading before they are detected and before an efficient antivirus appears, which can efficiently eliminate the viruses. An antivirus program appears only at a certain step of virus distribution, lagging behind the start of virus distribution by $h_0$ steps, i.e., at step k, k = h – $h_0$ (there is a delay).

The number of the antiviruses appearing at step (k + 1) (at step (h + 1) for viruses) is designated as $N_{k+1}$, and the number of viruses appearing at step k (step h for viruses) is denoted as $N_k$. In other words, $N_k$ will be equal to the number of computers, at which at k step an antivirus will be available, and $N_{k+1}$is equal to the number of computers, at which antivirus will be available at step (k + 1).

The number of computers infected at step (h + 1) can be defined as $P_{h+1}$, and the number of computers infected at step h can be indicated as $P_h$. The change in the number of infected computers is equal to the difference between the number of infections and the number of viruses destroyed at step (h + 1).

There are the following random events that form the complete system:

1. A computer is infected with a virus with probability of $\frac{P_h}{L}$.
2. There is an antivirus at the computer with probability of $\frac{N_k}{L}$.
3. There is neither a virus, nor an antivirus at the computer with probability of $\left\{1-\frac{P_h}{L}-\frac{N_k}{L}\right\}$.

The number of infections at step (h + 1) will be equal to $\xi\ P_h\left\{1-\frac{P_h}{L}-\frac{N_k}{L}\right\}$ as the infection of the already infected computer is not considered, and the computer where an antivirus is installed cannot be infected.

The number of viruses eliminated at step (h + 1) has to make up $P_h\frac{N_k}{L}$, where $\frac{N_k}{L}$ is the probability that at step (h+1) any of $P_h$ viruses existing at step h can encounter an antivirus. Thus

$$P_{h+1}\text{-}P_h = \xi\ P_h\left\{1-\frac{P_h}{L}-\frac{N_k}{L}\right\}-P_h\frac{N_k}{L} \tag{1}$$

The change in the number of computers where the antivirus is installed at step (k +1) is defined by $N_{k+1}$ – $N_k$ difference:

$$N_{k+1}\text{-}N_k = \xi\ P_h\left\{1-\frac{N_k}{L}\right\}, \tag{2}$$

where $\xi P_h$ implies that the antivirus is installed at step (h + 1) at those computers where a virus has been detected at step h, and $\left\{1-\frac{N_k}{L}\right\}$ means that the antivirus is installed only where it has not been present.

Since the duration of each step is equal to τ, the duration of the whole process t and number of steps h are interconnected by the following ratio of t=hτ and $t_0$ = $h_0$τ (k = h – $h_0$), where $t_0$ is the time when the antivirus springs into acting (its action lags behind the onset of viruses by the interval time value $t_0$).

Proceeding from the number of steps h and k to the process duration, we will receive:

$$P(t+\tau)-P(t)=\xi P(t)\left\{1-\frac{P(t)}{L}-\frac{N(t-t_0)}{L}\right\}- -P(t)\frac{N(t-t_0)}{L} \tag{3}$$

$$N(t-t_0+\tau)-N(t-t_0)=\xi P(t)\left\{1-\frac{N(t-t_0)}{L}\right\} \tag{4}$$

We will denote t-$t_0$ = y and, having decomposed (3) and (4) into a Taylor row, we will receive:

$$\tau\frac{dN(y)}{dy}+\frac{\tau^2}{2}\frac{d^2N(y)}{dy^2}+\cdots=\xi P(t)\left(1-\frac{N(y)}{L}\right) \tag{5}$$

$$\tau\frac{dP(t)}{dt}+\frac{\tau^2}{2}\frac{d^2P(t)}{dt^2}+\cdots= =\xi P(t)\left\{1-\frac{P(t)}{L}-\frac{N(t-t_0)}{L}\right\}- -P(t)\frac{N(t-t_0)}{L} \tag{6}$$

with an entry condition of N (y=0) = P ($t_0$), where y = t – $t_0$.

The equations (5) and (6) essentially differ from the system of equations used in the SIR model. The fundamental differences are:

- In the offered equation of infection (6), the decrease of viruses in the right part is determined not only by a share of nodes susceptible to infection $\left\{1-\frac{P(t)}{L}-\frac{N(t-t_0)}{L}\right\}$, but also by the product of the number of viruses and the probability of their encounter with $\frac{N}{L}$ antivirus, whereas the SIR model implies that the number of viruses decreases with the constant average speed of "immunization" per unit of time γ. Besides, the second derivative reduces the infection speed due to mutually reciprocal attacks (when we transfer it to the right member of the equation, a minus sign appears).
- The SIR model assumes that the speed of antivirus's emergence (the speed of disinfection) linearly depends on the number of available viruses. In the model offered (5), it depends on the probability of $\frac{N}{L}$ antivirus presence in the node and it is indirectly affected via $\frac{d^2N}{dt^2}$ on updating of the antivirus base. When transferring this member of the equation into the right part, a minus sign occurs, and the second derivative implies that the already available antivirus protection requires a base update, and instead of mailing over a network and installation of new antiviruses, they are just being updated.

This approach allows deducing the following differential equation that describes the kinetics of computer viruses epidemic development without protection by an antivirus (N (t-t$_0$) =0):

$$\frac{\mathrm{dP(t)}}{\mathrm{dt}} = \xi \mathrm{P(t)} \left(1 - \frac{\mathrm{P(t)}}{\mathrm{L}}\right) - \mathrm{O}\left(\frac{\mathrm{d^nP(t)}}{\mathrm{dt^n}}\right) \qquad (7)$$

The left member of (7) describes in general the speed of emergence of new infected computers or network nodes. The member of the (7) $\xi\mathrm{P(t)}\left(1\text{-}\frac{\mathrm{P(t)}}{\mathrm{L}}\right)$ describes the inception of new infected computers, i.e. the existence in (7) of just this summand implies that all copies of viruses penetrate only the computers that are not infected. Moreover, the member of $O\left(\frac{d^nP(t)}{dt^n}\right)$ view allows considering some part of the dispatched viruses to penetrate the already infected nodes (and thus reduce the infection since there is a minus sign before it).

Figure 7 presents the comparison of results of the empirical SI model and the model we offer, which is based on differential (7) that considers the derived changes of the second and higher order of the number of viruses over time. Curve 1 represents the SI model, and curve 2 depicts the offered model that takes into account the second derivative, curve 3 is *idem* for the third derivative, curve 4 is *idem* for the fourth derivative, and curve 5 is the same model taking into account the fifth derivative. All results are received for identical values of parameters (τ = 25, L = 200000, P$_0$ = 5 and ξ = 2).

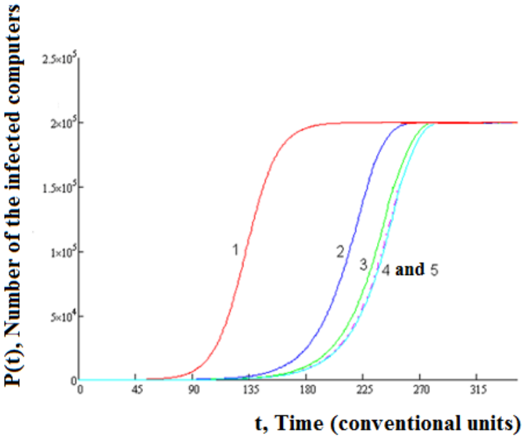


Figure 7. The comparison of SI model (curve 1) and the solution of the equation of the offered model.

Figure 8 presents the comparison of the offered model and the results of observation over development of epidemic of Code Red Worm [18] (the curve describing data is deduced using (7) and taking into account the summands of $O\left(\frac{d^nP(t)}{dt^n}\right)$ form, the dotted line represents experimental data). Figure 8 shows that the data observed and theoretical calculations coincide well with values of P$_0$ =1 (attack begins with one node), ξ =3 (the coefficient of reproduction equals 3, it is chosen to adjust the theoretical curve to the observed data), L=350000 (according to the observations presented in [16], the attacked network consisted of 350000 nodes), τ = 70 (duration of one step of the epidemic development equals 70 conventional units of time or 3.89 hours, it is chosen to adjust the theoretical curve to the observed data). Thus, the number of computers infected per hour is $\beta = \frac{\xi}{\tau} = 0{,}77$ . It coincides with an assessment of 0.7 to 1.8 nodes provided in [18] for random mailing.

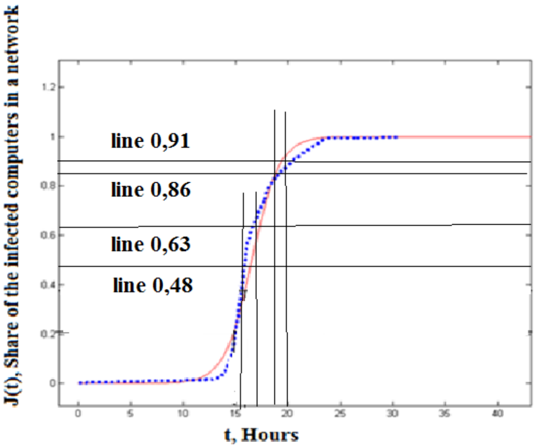


Figure 8. Comparison of observations over epidemic of Code Red worm (dotted line) and calculations according to the offered model (solid line).

In Figure 8, the horizontal lines show the allowance for the values of percolation thresholds ranging from 0.09 to 0.14 for a random network with the set of ways between nodes blocked by viruses (line1-0.09=0.91 and line 1-0.014=0.86). As seen in Figure 8, if the percolation threshold is to be considered as a criterion of reliability and operability of a computer network in general, then the network shut down from the normal operating mode will occur approximately in 19-20 hours after the beginning of infection, which allows taking necessary measures to eliminate epidemic consequences. This begs a question of why infection of computers continued. The answer is that a network consists not only of nodes that can be potentially infected, but also of nodes which cannot be any how infected due to the lack of vulnerabilities since they have another type of software. Yet, these nodes can transmit viruses through the network from infected to not infected nodes, remaining invulnerable. Besides, being infected, network nodes can continue carrying out functions on data transmission.

## IV. CONCLUSIONS

Open global computer networks and networks of data communication can be considered as the structures with random topology, and the processes running in such networks can be described by percolation theory.

During computer viruses epidemics distribution, data communication network nodes blocking can happen, as well as formation of clusters of such nodes. There are a number of blocked nodes at which all network entirely loses operational capacity (hitting the percolation threshold) in spite of the fact that a considerable part of nodes is still in an operational state.

For a random network, in the limit of infinitely large number of communications per node, it is enough to have the fraction of unblocked nodes equaling 0.09 relative to the total number so that there can arise a transferring chain of nodes and the network can solve the assigned information task. At an average of communications equaling 100, the percolation threshold will equal 0.094, and at 10 – 0.139. Thus, it is possible to consider that the fraction from 0.09 to 0.14 nodes keeping operational capacity allows providing overall operability of a network and its reliability. When we reduce the average of communications per node up to 2.5 – 3.5, the network keeps the general operational capacity for number of unblocked nodes from 0.52 to 0.37.

The smaller the average number of connections per node is, the less the blocking time of data communication network as a whole and attaining the percolation threshold turns out to be.

The size of a cluster of the blocked nodes depends on the average of communications and blocking probability. With the growth of an average of communications, at the fixed blocking probability, the size of cluster increases. A similar situation is observed with a large average of communications per node of a random network.

Practical recommendations for protection of any data communication networks against threats of virus attacks are essentially the following. When using the equitype equipment and software to create networks of data communication with a very large number of communications per node, its share should not exceed 86% - 91%. It will allow keeping operability of all network as a whole during epidemics spread of multivector viruses capable of deploying for their penetration not one, but the whole set of vulnerabilities, since it increases the probability that 9% -14% of the employed equipment and types of software will be impregnable. However, in reality the average of communications per node of network varies from 2.5 to 3.5, which yields the percolation threshold ranging from 0.52 to 0.37. Thus, for the real computer information networks the fraction of the employed equitype equipment and software types should be strictly within the limit from 48% to 63% (1-0.52=0.48 and 1-0.37=0.63).

Describing the distribution dynamics of computer viruses and using the differential equations of the second and higher order to account for changes in the number of the infected nodes over time, the mathematical model enables allowances for mailing copies to already infected addresses. It will also be significantly better coordinated with the results of observation over computer viruses epidemics in the Internet, than the existing SI and SIR models are.

In the future we will consider probabilistic schemes of transitions between statuses of congestion of data transmission networks. Such formalization is able to receive the differential equation of second order (like Kolmogorov's equation) which modeling stochastic dynamics of change status of congestion a network and to connect them with the results received from the percolation models.

## REFERENCES

[1] J. Nazario, "Defense and Detection Strategies against Internet Worms"; Artech House, 2004, ISBN: 1580535372.

[2] J. Leveille, "Epidemic Spreading in Technological Networks", Information Infrastructure Laboratory HP Laboratories Bristol, 2002, available at http://www.hpl.hp.com/techreports/2002/HPL-2002-287.pdf

[3] R. Pastor-Satorras and A. Vespignani, "Epidemic dynamics and endemic states in complex networks", Physical Review E, vol. 63, 2001, pp. 0661171 – 0661178.

[4] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in finite scale-free networks", Physical Review E, vol. 65, 2001, pp. 0351081 – 0351084.

[5] R. Pastor-Satorras and A. Vespignani,"Epidemic spreading in scale-free networks", Physical Review Letters, vol. 86, 2001, pp. 3200–3203.

[6] R. Pastor-Satorras and A. Vespignani, "Epidemics and Immunization in Scale-Free Networks", Wiley-VCH, S. Bornholdt and H. G. Schuster (eds.) Handbook of Graphs and Networks: From the Genome to the Internet, 2005, DOI:10.1002/3527602755.ch5.

[7] R. Pastor-Satorras and A. Vespignani,"Immunization of complex networks", Physical Review E, vol. 65, 2002, pp. 036104.

[8] A. Fekete, G. Vattay and L. Kocarev, “Traffic Dynamics in Scale-Free Networks”, Complexus, vol. 3, 2006, pp. 97–107, DOI: 10.1159/000094192.

[9] Zhi-Xi Wu, G. Peng, Wing-Ming Wong and Kai-Hau Yeung, “Improved routing strategies for data traffic in scale-free networks”, Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, 2008, P11002, DOI:10.1088/1742-5468/2008/11/P11002.

[10] S. Boccaletti, D.-U. Hwang and V. Latora,“Growing hierarchical scale-free networks by means of nonhierarchical processes”, International Journal of Bifurcation and Chaos, Vol. 17, No. 7,2007, pp. 2447–2452, DOI:10.1142/S0218127407018518.

[11] G. Grimmett, "Percolation and disordered systems, in Lectures in Probability Theory and Statistics", Ecole d'Eté de Probabilités de Saint-Flour XXVI-1996, Springer Lecture Notes in Math. no. 1665, ed. P. Bernard, 1997, ISBN978-3-540-63190-3.

[12] G. Grimmet, "Percolation", Springer-Verlag, 1999, ISBN978-3-540-64902-1.

[13] M. B. Isichenko, "Percolation, statistical topography, and transport in random media", Rev. Mod. Phys., vol. 64, 1992, pp.961-1043, http://dx.doi.org/10.1103/RevModPhys.64.961.

[14] M. Sahimi, "Applications of Percolation Theory", Taylor & Francis, 1992, ISBN 0748400761.

[15] V.K.S. Shante and S. Kirkpatric, "An Introduction to Percolation Theory", Advances in Physics, vol. 85, 1971, pp. 325-357, DOI:10.1080/00018737100101261.

[16] W.W. Wilcke, R.B. Garner, H. Huels, “Percolation in dense storage arrays”, Physica A: Statistical Mechanics and its Applications, Vol. 314(1), 2002, pp. 220-229, DOI:10.1016/S0378-4371(02)01153-6.

[17] C. C. Zou, D. Towsley, W. Gong, “On the performance of Internet worm scanning strategies”, Performance Evaluation vol. 63, 2006, pp. 700–723, DOI:10.1016/j.peva.2005.07.032.

[18] C. C. Zou, W. Gong, D. Towsley, “Code Red Worm Propagation Modeling and Analysis”, 9th ACM Symposium on Computer and Communication Security, 2002, pp.138 – 147.

# Terminal Virtualization for Mobile Services

Tao Zheng, Song Dong
Orange Labs International Center
Beijing, China
e-mail: {tao.zheng, song.dong}@orange.com

***Abstract*–Terminal virtualization focuses on applying Information Technology (IT) virtualization technology to the terminals to realize the full or parts of terminal functions extension or migration to other devices on the network, such as resource reducing, information sharing, data synchronization, etc. It is becoming increasingly clear that more and more features of terminal virtualization and mobile computing on the edge will be used in practice. However, some issues are raised with terminal virtualization, such as security, privacy, Quality of Service (QoS), efficient transmission, computation/functions offloading management, etc. In this paper, after analyzing above issues, a mobile terminal virtualization framework is proposed and considered to be implemented in terminal Operating System (OS) and transparent to users.**

***Keywords-terminal virtualization; mobile cloud computing; computation offloading; QoS***

## I. Introduction

With the explosive growth of mobile terminals in recent years, user preferences have shifted from traditional cell phones and laptops to smartphones and tablets. In recent years, there are abundant applications in various categories, such as entertainment, health, games, business, social networking, travel and news, running at mobile terminals. The burden of computation on the terminals has been raised rapidly and more functions and sensors are required to be applied to them. Mobile cloud computing and terminal virtualization are proposed to handle these issues, which are able to provide tools to the user when and where it is needed irrespective of user movement, hence supporting location independence. Indeed, “mobility” is one of the characteristics of a pervasive computing environment where the user is able to continue ones work seamlessly regardless of the movement.

Advances in the portability and capability of mobile terminals, together with widespread Long Term Evolution (LTE) networks and WiFi access, have brought rich mobile application experiences to end users. Undoubtedly, mobile broadband terminals, such as smart phones, tablets, wireless dongles and some data-intensive apps have been an exponential increase in mobile Internet Protocol (IP) data usage, which will used up the mobile bandwidth. The demand for ubiquitous access to a wealth of media content and services will continue to increase, as indicated in a report by Cisco [1]: the Compound Average Growth Rate (CAGR) of global IP traffic from mobile terminals is 61% from 2013 to 2018, which is triple CAGR from fixed Internet.

In addition, the resource-constrained mobile terminals, especially with limited battery life, have been a barrier to the improvements of mobile applications and services. While new smart phones with bigger screens, faster Central Processing Units (CPUs), and larger storage are launched continually, and the bandwidth of wireless networks has increased hundreds of times in just a few years, the development of batteries has lagged far behind the development of other components in mobile terminals. In fact, faster CPUs, larger displays and multimedia applications consume more battery energy. The limitations of computation resources and sensors are other stumbling blocks for services development. Mobile cloud computing and terminal virtualization can help to resolve this issue.

Mobile cloud computing and terminal virtualization have been the leading technology trends in recent years. The increasing usage of mobile computing is evident in the study by Juniper Research, which states that the consumer and enterprise market for cloud-based mobile applications is expected to rise to $9.5 billion by 2014 [2]. Mobile cloud computing/terminal virtualization is introduced to resolve the conflicts mentioned above, in which the cloud serves as a powerful complement to resource-constrained mobile terminals. Rather than executing all computational and data operations locally, mobile cloud computing/terminal virtualization takes advantage of the abundant resources in cloud platforms to gather, store, and process data for mobile terminals. Many popular mobile applications have actually employed cloud computing to provide enhanced services. More innovative cloud-based mobile applications like healthcare monitoring and massively multiplayer online mobile games are also under development.

The objective of the paper is to introduce the concept of terminal virtualization and study the related issues and research status. On this basis, a proposal terminal virtualization framework is finally presented.

This paper is organized as follows. In Section 2, we introduce the concept and current status of terminal virtualization. In Section 3, capabilities and functions extension are analyzed. In Section 4, a terminal virtualization framework for mobile networks is presented. Some implementing issues about this framework are discussed in Section 5. Finally, Section 6 summarizes the conclusions.

## II. Terminal Virtualization

Terminal virtualization helps to relief the local resource-constrained problem through offloading some tasks to the cloud and utilizing capabilities and functions in the cloud. First, the scope of terminal virtualization needs to be clarified.

### *A. The scope of terminal virtualization*

From the points of view of virtualization, mobile cloud computing can be as a kind of terminal virtualization scenario. There are two scenarios as following:

- Full Virtualization Scenario

The requirement for full terminal virtualization mainly comes from some enterprises. In these enterprises, employees are buying their own terminals and want to connect to the enterprise network so that they can do their work with greater flexibility. However, the employees also don't want to give up user experience and freedom at the cost of complex IT security policies. In order to achieve this goal, terminal virtualization is becoming a very attractive choice because it offers flexibility and addresses the concerns over privacy of personal data while also delivering the security requirements of the enterprise. On the other side of the ecosystem, the terminal makers and carriers will benefit from terminal virtualization because they are able to more easily replicate the features found in various terminals and also deliver more features at a lower cost.

Full terminal virtualization is not an ordinary schema for public mobile customers. In general way, the terminal is sold with a pre-determined OS and customers can use services based on this OS.

- Partial Virtualization Scenario

Broadly speaking, mobile cloud computing can be as one kind of partial terminal virtualization, a part of terminal computation powers and functions can be virtualized into the remote networked cloud. Terminals can get local experience through running remote apps or some information located in the remote cloud.

This scenario is more practiced and popular in present. Some applications employ this method to add extending functions or improve user experience. Even cloud phone appears and is deeply merged with networking services for user convenience. In this paper, we mainly focus on the partial virtualization scenario.

### *B. Drivers and Benefits of Terminal Virtualization*

Terminal virtualization facilitated the fusion of mobile terminal and cloud service that provides a platform wherein some computing, storing and data abstraction tasks are performed by the cloud and mobile terminal simply seeks an access to them. Following shows the drivers and benefits of terminal virtualization:

- Limitless Storage Space

Now, instead of memory cards for more space, the cloud storage can provide limitless space for applications, even with the help of terminal virtualization framework/middleware they don't need to care about the location of the storage.

- Improved Processing Facility

The price of a mobile terminal is largely dependent on its CPU's speed and performance. With the help of terminal virtualization, all the extensive and complex processing is done at the cloud level. The vital computations, encryption and decryption, everything can be handled by the cloud thereby enhancing the mobile terminal's performance.

- Save Radio Access Network (RAN)/Access Bandwidth & Resources

With the tremendous increase in mobile bandwidth consumers and user's throughput, RAN/access resources have become more valuable than before. When some functions and computation tasks are offloaded into cloud, the result instead of the original metrical is sent to the terminal, so the RAN/access bandwidth can be saved for other use.

- Enhanced Battery Life

Terminal virtualization lends a very strong helping hand to battery life of terminals. With most of the processing handled by the cloud, the battery life is enhanced, thereby making the most optimum use of the remaining recharge cycles.

- Improved User Experience

The above mentioned features will improve the end-user experience substantially, especially the experience from low-end terminals.

- Economic Factors

For the consumers, terminal virtualization can bring some new functions and improved capabilities to the old terminal without spending one penny. For operators, the benefits come from saved network resources and flexible service deployment by terminal virtualization.

- Reserving for Upcoming Technologies

Terminal virtualization is adapt with the tremendous pace of developing technology and works most efficiently with the upgrades. Through separating the implementation from the function body, upcoming technologies can be easy to be introduced to the terminals.

### *C. Challenges*

In this section, we discuss that the issues have not been sufficiently solved in terminal virtualization.

- Energy-efficient Transmission

Wireless networks are stochastic in nature: not only the availability and network capacity of access points vary from place to place, but the downlink and uplink bandwidth also fluctuates due to weather, building/geographical shields, terminal mobility, and so on. Measurement studies [3] show that the energy consumption for transmitting a fixed amount of data is inversely proportional to the available bandwidth.

Computation/Data offloading can save energy only if heavy computation is needed and a relatively small amount of data has to be transferred. Energy efficiency can be substantially improved if the cloud stores the data required for computation, reducing data transmission overhead. Bandwidth allocation and admission control mechanisms in cellular base stations and access points may guarantee network connectivity to a certain extent, but cannot eliminate the stochastic nature of wireless links. An alternative approach is to dynamically adjust application partitioning between the cloud and mobile terminals according to network conditions, although it is challenging to quickly and accurately estimate the network connectivity with low overhead.

Energy-efficient transmission is also critical when exploiting the cloud to extend the capabilities of mobile terminals. Frequent transmissions in bad connectivity will overly consume energy, making the extended capabilities unattractive, as battery life is always the top concern of mobile users. A solution called eTime [4] is to adaptively seize the timing opportunity when network connectivity is good to pre-fetch frequently used data while deferring delay-tolerant data.

- Security

There are several aspects of terminal virtualization security, including antivirus, authentication, data protection, and digital rights management. Security vulnerability can cause serious problems, including property damage, cloud vendor economic loss, and user distrust. Since mobile terminals are resource-constrained, locally executed antivirus software can hardly protect them from threats efficiently. A current solution is to offload the threat detection functionality to the cloud. Nevertheless, since a pure cloud antivirus relies on cloud resources, it is difficult to deal with malware that can block the terminal's Internet connection.

Besides, authentication is critical for access to sensitive information, such as bank accounts and confidential files. With constrained text input on mobile terminals, users tend to use simple passwords, making mobile applications more vulnerable to authentication threats. To solve this issue, Chow et al. [5] builds up an authorization platform where users are identified by their habits (e.g., calling patterns, location information, and web access). The platform routinely records user behavior information. When a server receives an authorization request, it redirects the request to an authorization engine, which uses the aggregated behavior information and an authorization policy to decide whether to accept the request or not.

- Privacy

Since mobile terminals are usually personal items, privacy must be considered when leveraging the cloud to store and process their confidential data remotely.

A secure data processing framework [6] can be used into terminal virtualization, in which critical data are protected by the unique encryption key generated from the user's trusted authority and stored in an area isolated from the public domain. Even when storage is breached in the cloud, unauthorized parties including the cloud vendor cannot obtain the private data.

Another particular privacy issue for mobile users is the leakage of personal location information in location-based services. To address the issue, a method called "location cloaking" [7] makes user location data slightly imprecise before submitting them to the cloud. But the imprecise data sometimes cannot provide relevant or satisfactory results to users in certain applications. Therefore, location cloaking should be adaptively tuned to balance the trade-off between privacy and result accuracy.

- Real-time Requirements and Service QoS

When terminal virtualization and mobile computing are applied, QoS will become more important. How to guarantee the related data or stream to be transmitted in time determinates the services' failure or success.

While different applications offer different functionality to end users, the primary service Key Quality Indicators (KQIs) across the application's customer facing service boundary for end users of applications generally include service availability, service latency, service reliability, service accessibility, service throughput, and application specific service quality measurements.

## III. Computation Offloading & Functions Extension

Terminal virtualization enables enhanced mobile experiences that were previously impossible on resource-constrained and function-constrained mobile terminals. Many commercial mobile applications use the cloud to bring about rich features. They usually employ a client-server framework that consists of two parts, which run on the mobile terminal and the cloud, respectively. Essentially, cloud computing helps extend the capabilities and functions of mobile terminals in some aspects.

### *A. Capabilities & Functions extension*

Through terminal virtualization, the capabilities and functions can be reallocated between terminal and cloud, as shown in the following examples:

- Computation-intensive Task

At present, many applications nowadays support speech/picture/video recognition. The models for recognition and high-quality synthesis must be trained with millions of samples in thousands of examples. This computation-intensive task is infeasible on a mobile terminal and should be offloaded to the cloud.

- Remote Sensors/Inductors

Because of the limitation from terminal itself (low-end model lacking some sensors or inductors) or other conditions (e.g., distance exceeding the maximum length of sensors), some services cannot work well. However, these services can work through getting and storing related information from mobile cloud platform. From the terminal point of view, its functions are extended.

- Application Portability

It allows for the rapid transfer of applications (which may occur on the fly), providing the freedom to optimize, without the constraints of the location and required resources of the virtual appliances. The precise but extensible definition of the services provided by the application platform is the key to ensuring application portability.

### B. Computation offloading Decision

To overcome resource constraints on mobile terminals, a general idea is to offload parts of resource-intensive tasks to the cloud (centralized server or other peers). Since execution in the cloud is considerably faster than that on mobile terminals, it is worth shipping code and data to the cloud and back to prolong the battery life and speed up the application. This offloading procedure is illustrated in Fig. 1. Several technologies to realize the runtime environment in the cloud, the major differences between offloading techniques lie in the offloading unit and partitioning strategies.

- Client–Server Communication Mechanism

In the Client–Server Communication, process communication is done across the mobile terminal and cloud server via protocols, such as Remote Procedure Calls (RPC), Remote Method Invocation (RMI) and Sockets. Both RPC and RMI have well supported APIs and are considered stable by developers. However, offloading through these two methods means that services need to have been pre-installed in the participating terminals.

Spectra [8] and Chroma [9] are the examples of systems that use pre-installed services reachable via RPC to offload computation. Hyrax [10] has been presented for Android smartphone applications which are distributed both in terms of data and computation based on Hadoop ported to the Android platform. Another framework based on Hadoop is presented by in [11], for a virtual mobile cloud focusing on common goals in which mobile terminal are considered as resource providers. Cuckoo [12] presents a system to offload mobile terminal applications onto a cloud using a Java stub/proxy model. The Mobile Message Passing Interface (MMPI) framework [13] is a mobile version of the standard Message Passing Interface (MPI) over Bluetooth where mobile terminals function as fellow resource providers.

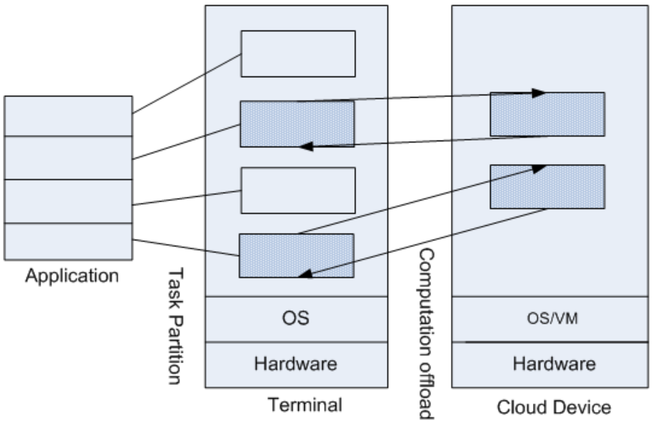


Figure 1. The procedure of computation offloading

- Mobile Agent

Scavenger [14] is another framework that employs cyber-foraging using WiFi for connectivity, and uses a mobile code approach to partition and distribute jobs. Using its framework, it is possible for a mobile terminal to offload to one or more agents and its tests show that running the application on multiples in parallel is more efficient in terms of performance. However, the fault tolerance mechanism does not be discussed and since its method is strictly about offloading on agents and not sharing, it is not really dynamic. Also its agents are all desktops and it is unclear if Scavenger is too heavy to run on mobile phones.

- Virtualization/Virtual Machine (VM) Migration

The execution can’t be stopped when transferring the memory image of a VM from a source terminal to the destination server [15]. In such a live migration, the memory pages of the VM are pre-copied without interrupting the OS or any of its applications, thereby providing a seamless migration. However, VM migration is somewhat time-consuming and the workload could prove to be heavy for mobile terminals.

VM migration is used by a majority of frameworks, including Cloudlets [16], Maui [17], CloneCloud [18], and MobiCloud [19]. Virtualization greatly reduces the burden on the programmer, since very little or no rewriting of applications is required. However, full virtualization with automatic partitioning is unlikely to produce the same fine grained optimizations as that of hand coded applications, although rewriting each and every application for code offload is also not practical. Maui actually does not rely on pure VM migration as done in CloneCloud and Cloudlets, but uses a combination of VM migration and programmatic partitioning. However, in cases where the mobile terminal user is within range of an agent terminal for a few minutes, using VM migration may prove to be too heavyweight, as is pointed out by Kristensen [14] which uses mobile agents in light of its suitability in a dynamic mobile environment.

### C. Applications

The following lists the current applications using terminal virtualization concept, from functions extension to complex computing tasks.

- Mobile Cloud Phone

Mobile cloud phone differs from other smart phones in that it doesn’t need to download and store apps and content on the phone; it instead accesses personal information and runs programs stored on remote network servers, via the cloud.

YunOS 3.0 [20], developed by Alibaba, debuts officially with cloud-based service for movie, taxi and other reservations on October 20th, 2014. It comes with the brand new service Cloud Card, which runs entirely in the cloud and offers the user the option to select movie tickets, taxi services and more.

- Cloud Storage and Video Adaption

Through terminal virtualization, some part of data which is stored in the cloud instead of being stored on the terminal can be treated as local data. And video stored in cloud platform can be adapted to appropriate format and code streaming fitting for the terminal when the terminal requests this video.

- Image and Natural Language Processing

For this kind of applications, the complex computation jobs which are difficult for local operating OS should be offloaded to the cloud platform and the mobile terminal just holds some interface functions. Image and voice recognition, search, adaption, natural language translation, and Artificial Intelligence (AI) machine conversation, etc., which belong to this kind of applications, can be implemented on some low-end phones with the help of computation offloading.

- Augmented Reality (AR)

Algorithms in augmented reality are mostly resource and computation-intensive, posing challenges to resource-poor mobile devices. These applications can integrate the power of the cloud to handle complex processing of augmented reality tasks. Specifically, data streams of the sensors on a mobile device can be directed to the cloud for processing, and the processed data streams are then redirected back to the device. It should be noted that AR applications demand low latency to provide a life-like experience.

## IV. Proposed Framework for Mobile Services

This section proposes a mobile terminal virtualization framework based-on mobile OS. The framework locates in the middleware layer between OS kernel and applications, as shown in Fig. 2.

### A. *The framework overview*

The framework illustrated in Fig. 3 is composed of four processing modules: application virtualization module, computation virtualization module, storage virtualization module and network virtualization module, and a management module.

In the framework, processing modules are in charge of receiving and responding the callings from applications to OS. Management module is used to manage the framework, including security management, configuration management, network and cloud service monitoring, etc.

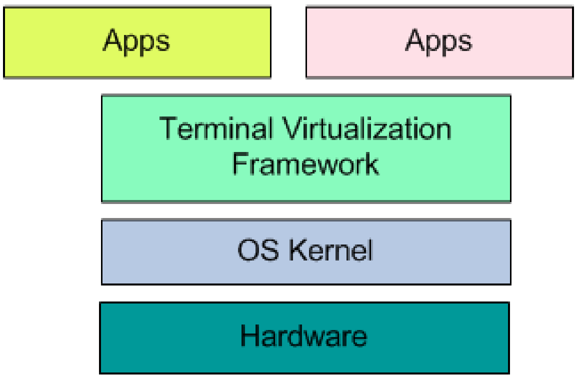


Figure 2. Hierarchical structure of the framework

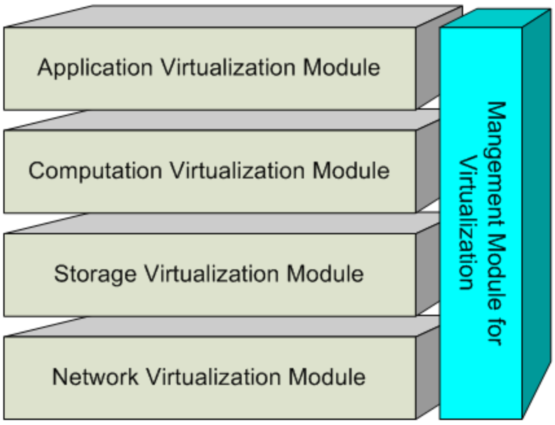

Application Virtualization Module
Computation Virtualization Module
Storage Virtualization Module
Network Virtualization Module
Mangement Module for Virtualization

Figure 3. The overview of the framework

### B. *Component Modules*

As shown in Fig. 4, the processing modules are able to choose the best method to process the calling according to the current status of mobile network bandwidth, local resources, terminals hardware limitation and remote cloud resources. Meanwhile, the framework provides local calling responses to the applications and shields the actual calling responses.

- Application Virtualization Module

This module is in charge of processing the functions extension of applications. When the application accesses the terminal's hardware, for example one kind of sensor, this module will check if it is available. If not, this module is responsible for finding a same remote available sensor in the cloud to satisfy the application's demand and providing the response with the result from the remote sensor to the application.

- Computation Virtualization Module

This module takes charge of monitoring the terminal's computation resource and analyzing the computation request from applications. When the computation resource is constrained (for example, the CPU usage is more than 85%) or some applications with sophisticated computing power are run (for example, virtual reality service, language processing, etc., it can be configured in advanced), this module will offload some computation tasks to the remote cloud server and provide the processing result to the applications.

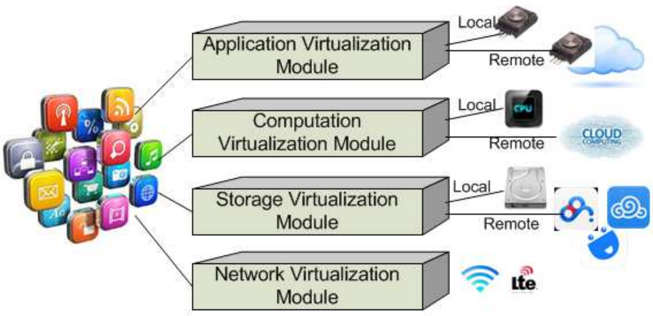

Application Virtualization Module
Local
Remote
Computation Virtualization Module
Local
Remote
CPU
CLOUD
Storage Virtualization Module
Local
Remote
Network Virtualization Module
Lte

Figure 4. The functions of processing modules

- Storage Virtualization Module

This module is in charge of monitoring the terminal's storage resource and providing the remote cloud storage to applications. Through this module, the local applications can use cloud storage providing by different Service Providers (SPs), such as Baidu, Tencent, Huawei, etc., as using a local storage. At the same time, this module monitors the speed and status of the remote cloud storages and provides the best one to applications.

- Network Virtualization Module

This module is in charge of monitoring the terminal's network status and providing the best one or binding different network accesses to increase the data throughput according to the applications' demand.

- Management Module

The management module is responsible for managing the framework.

The security function including network security and resources security is an important function in this module. Other management functions include all kinds of configuration management, for example, some resources and offloading thresholds, and remote resource monitoring, for example, all kinds of cloud services, network status.

## V. Considerations for Framework Implemention

We are implementing an early-phase prototype based-on Android OS according to the proposed framework. The implemented modules include network virtualization module and storage virtualization module, which are relatively easier to be implemented than other three modules in the framework.

The network virtualization module employed a method [21] to implement the network access independence, for example, using multiple access paths simultaneously, switching between access paths according to the current network environment, and recovering the access path automatically. A local proxy in terminal and a remote proxy in network cooperate to implement the functions of network virtualization module. And the applications are able to automatically adapt the change of network and not affected by it.

The storage virtualization module added online storage services to the local storage as a directory, which can be accessed by the applications as a local one. When the directory is accessed, the storage virtualization module will automatically exchange the data with the online storages. Baidu and Huawei online storage services are currently supported and chosen by the module.

To implement application virtualization, some operating system calls to local hardware need to be intercepted and rewritten. The functions to access online hardware resource will be added into the application virtualization module.

In the computation virtualization module, we plan to take different approaches according to the type of tasks. For example, for the tasks requiring sophisticated computing power defined in advance, RPC/RMI method will be used; for the independent tasks undefined in advance, virtual machine migration will be used.

To implement the security function in the management module, the data communication between the terminal and cloud platform is encrypted. In addition, security function also helps protect the terminal and remote resources from being abused by applications on the terminal.

## VI. Conclusion and Future Work

In this paper, we analyze some aspects of current terminal virtualization, highlight the motivation for it, and present its functions, applications and some challenges. Terminal virtualization has overlapped with other areas, such as mobile peer-to-peer computing, application partitioning, and context-aware computing, but it still has its own unique challenges. These are still a long way to go in terminal virtualization.

Because more and more cloud resources can be made available to the mobile terminal (via the mobile cloud facility), we proposed a terminal virtualization framework for mobile services. In this framework, four processing modules and one management module are employed to handle the resource requests from apps and shield the details for accessing cloud services. In the future work, we consider completing the prototype of this framework and analyzing its performance.

## References

[1] Cisco Visual Networking Index: Forecast and Methodology, 2013–2018, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html, March 2015.

[2] "Mobile Cloud Applications & Services", Juniper Research, 2010.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: a Measurement Study and Implications for Network Applications," Proc. ACM IMC, 2009, pp. 280-293.

[4] P. Shu, F Liu, H. Jin, M. Chen, F. Wen, and Y. Qu, "eTime: Energy-Efficient Transmission between Cloud and Mobile Devices," Proc. IEEE INFOCOM, 2013, pp. 195-199.

[5] R. Chow et al., "Authentication in the Clouds: a Framework and Its Application to Mobile Users," Proc. ACM Cloud Computing Security Wksp. 2010, pp. 1-6.

[6] D. Huang, Z.Zhou, L. Xu, T. Xing, and Y. Zhong, "Secure Data Processing Framework for Mobile Cloud Computing," Proc. IEEE INFOCOM, 2011, pp. 614-618.

[7] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving User Location Privacy in Mobile Data Management Infrastructures," Proc. Wksp. Privacy Enhancing Technologies, 2006, pp. 393-412.

[8] J. Flinn, S. Park, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing", Proc. IEEE ICDCS 2002, pp. 217-226.

[9] R. Balan, M. Satayanarayanan, S. Park, and T. Okoshi, "Tactics-based remote execution for mobile computing", Proc. ACM Mobisys, 2003, pp. 273-286.

[10] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce", Masters Thesis, Carnegie Mellon University, 2009.

[11] G. Huerta-Canepa, and D. Lee, "A virtual cloud computing provider for mobile devices", Proc. ACM MCS 2010, article No. 6.

[12] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a computation offloading framework for smartphones", Proc. ACM Mobisys, 2010, pp. 59-79.

ISBN: 978-1-61208-404-6

[13] D. C. Doolan, S. Tabirca, and L.T. Yang, “Mmpi a message passing interface for the mobile environment”, Proc. ACM Mobisys, 2008, pp. 317-321.

[14] M. Kristensen, “Scavenger: transparent development of efficient cyber foraging applications”, Proc. IEEE PerCom, 2010, pp. 217-226.

[15] C. Clark, et al., “Live migration of virtual machines”, Proc. of the 2nd conference on Symposium on Networked Systems Design & Implementation, USENIX Association, 2005, vol 2, pp. 273-286.

[16] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM –based Cloudlets in Mobile Computing,”, IEEE Pervasive Computing, vol. 8, no. 4, 2009.

[17] E. Cuervo et al., “Maui: Making Smartphones Last Longer with Code Offload,” Proc. ACM MobiSys, 2010, pp. 49-62.

[18] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: Elastic Execution Between Mobile Device and Cloud,” Proc. ACM EuroSys, 2011, pp. 301-314.

[19] D. Huang, X. Zhang, M. Kang, and J. Luo, “MobiCloud: Building Secure Cloud Framework for Mobile Computing and Communication,” Proc. IEEE SOSE, 2010, pp. 27-34.

[20] http://www.yunos.com, YunOS, March 2015.

[21] T. Zheng, and D. Gu, “Traffic Offloading Improvements in Mobile Networks”, ICNS 2014, pp. 116-121.

# Internet of Things and OPC UA

Ravish Kumar
ABB Corporate Research
Bangalore, India
e-mail: ravish.kumar@in.abb.com

Arijit Kumar Bose
ABB Corporate Research
Bangalore, India
e-mail: arijit.bose@in.abb.com

***Abstract*— Internet of Things (IoT) has become very popular due to its envisioned capability. Industry, Health care and Utility sectors are working actively to take advantage of the benefits that the IoT infrastructure can offer. However, it will take a long time until a completely developed IoT infrastructure is in place. Since this requires a large scale technology restructuring, many challenges need to be addressed. Certainly, IoT infrastructure development would happen step by step and it will be slowly accepted by the users, particularly from the industrial sector. In addition, some provision is also required to connect the existing automation system with the IoT infrastructure. In order to facilitate this connection, a bridging technology is required. In this paper, we describe how the industry proven OPC UA technology can be used for connecting the existing automation system with an IoT infrastructure. Furthermore, we also analyze the OPC Unified Architecture (OPC UA) security model from an IoT perspective and highlight the required improvements.**

***Keyword-Internet of Things; OPC UA; Security***

## I. Introduction

Technology analysts and visionaries have defined Internet of Things (IoT) [5] as a network of physical objects which can be accessed through the Internet. These objects contain embedded technology to interact with the external environment. The objective of IoT is to mix the physical world and the information world. As a result, it will create an environment where one machine can communicate directly with other machines without much manual intervention. IoT has a great potential to influence Industrial application. It will create new ways of organizing processes and information flow across industrial production. By connecting different machines, field devices, production units, transportation information and goods data seamlessly to an IoT infrastructure, a smart production system can be created with more flexibility, resourceful and faster production. Such a smart production system will leverage to incorporate last minute changes in the production cycle and will also possess the ability to respond flexibly to disruptions and failures on behalf of suppliers and other external factors. In addition, the smart system will also induce the capability to respond rapidly to dynamic businesses and engineering processes, thereby facilitating dynamic changes in the production when needed.

For creating an IoT infrastructure, coordination and cooperation between the new generation technologies and existing proven technologies must be considered. Most of the existing industrial automation systems are based on the concept of a Distributed Control System (DCS) [1], which works in isolation and cannot be directly operated from the Internet. Due to this limitation, it has reduced flexibility to handle dynamic business changes in an industrial plant. Therefore, to evolve a DCS architecture to handle dynamic business changes, a new generation of technologies coming traditionally from an Internet world should be considered."

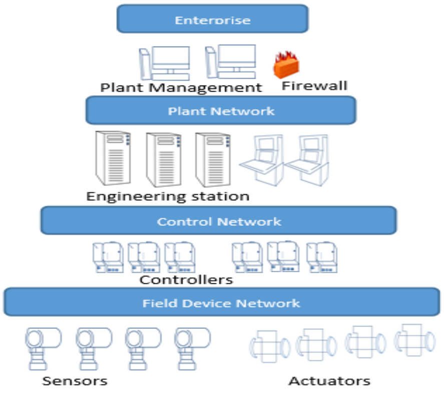


Figure 1. Distributed Control System Architecture.

DCS has different layers for handling various operations. Fig. 1 shows the hierarchy of a typical DCS based industrial automation system. Typically, there are four layers in an automation system. These layers are Enterprise Network layer, Plant Network Layer, Control Network Layer and Field Device Network Layer. All the layers are specialized to perform specific kinds of operations. For example, the Control Network Layer is responsible for the execution of control tasks of the plant process.

Currently, the industrial automation system works in isolation with other entities such as Enterprise Resource Planning (ERP) [2] and Manufacturing Execution System (MES) [3]. ERP is a business management software application, which is used for product planning, inventory and suppliers' management, shipping and payment, etc. MES is a software application, which is used for accessing current conditions of plant processes for resource optimization and decision making. Because they work in isolation, the existing industrial automation systems are not able to handle dynamic processes and to incorporate last minute changes into the process flow. Currently, incorporating any small updates in the production life cycle is expensive because multiple changes and synchronizations need to be done in different places. Connecting the industrial automation system with other entities such as ERP, MES, etc. over the IoT infrastructure produces a whole automation system capable of handling dynamic business and engineering processes.

In this paper, we will investigate the solution for enhancing the existing industrial automation system to leverage benefits from dynamic business and engineering process. This is done by enabling connectivity to an IoT infrastructure by using the industry proven OPC Unified Architecture (OPC) technology [4]. We will also analyze the security framework of OPC UA from an IoT perspective and will be highlighting the required improvements.

The rest of the paper is organized as follows. In Section II, we provide the background and related work of the IoT. In Section III, we provide an overview of the existing OPC UA standard, followed by describing how OPC UA can be used for enhancing the existing industrial automation system to connect with an IoT infrastructure. In Section IV, we provide an overview of the OPC UA security model. In Section V, we analyze the security of OPC UA from an IoT perspective and highlight the required improvements. In Section VI, we provide a conclusion to our work and describe the possible future work.

## II. Related Work

IoT is basically a convergence of multiple technologies such as Radio-frequency identification (RFID), sensor technology, Internet, wireless, cloud computing, etc. All of these technologies contribute to enable an IoT infrastructure. The term IoT was first introduced by Kelvis Ashton in the year 1999. His initial idea was to empower computers to gather information on their own, so that computers can see, hear and smell the world by themselves [5]. But, in today’s scenario, it is not just limited to empowering computer to gather information only. Now, it is considered as a communication infrastructure for exchanging information among the things around the globe [6]. Things, in the IoT, refer to a wide variety of devices. The applicability of IoT is not limited to one domain [7][8]. It is suitable for various application sectors like industrial domain, health care, utility, etc. Tan et al. [9] described IoT as future Internet for establishing communication not only between human to human, but expanding to human to machine, and machine to machine. Chen [10] discusses the overview of new paradigm along with different challenges and opportunities. Imtiaz et al. [11] investigated the OPC-UA as a middleware solution for resource-limited devices. To handle an enormous volume of IoT data, Copie et al. [12] highlighted how IoT data can be stored in a cloud database. The Industry 4.0 [13] revolution has been envisioned based on IoT and Cyber Physical System (CPS). Perera et al. [14] examined a variety of popular and innovative IoT solutions in terms of context-ware technology perspective and evaluated them on a framework that they built around well-known context aware computing theories. Singh [15] has presented an efficient hierarchical identification mapping server for identification and location of connected things in the IoT infrastructure for enabling global mobility and scalability. Ungurean et al. [16] discussed an IoT architecture based on the OPC.NET [17] technology. OPC.NET is tightly coupled with Microsoft platform. Because of this platform dependency, freedom of platform independency cannot be achieved in true sense. Furthermore, Ungurean et al. [16] have not discussed the security aspects of OPC.NET from the perspective of an IoT infrastructure.

Keoh et al. [18] provide some salient security enhancements that are required in the emerging IoT protocols like security enhancements for Constrained Application Protocol (CoAP), Datagram Transport Layer Security (DTLS), etc. Sajjad et al. [19] discussed on the security enhancements of IEEE 802.15.4 Media Access Control (MAC) in the context of IoT. However, based on the prior art, we did not find many studies that explain the security enhancements for OPC UA from an IoT perspective.

## III. OPC Unified Architecure

OPC UA [4] is an Industry proven standard for exchanging industrial data. It provides a framework for safe and reliable communication among the different industrial devices and applications. This standard is developed with close cooperation with manufacturers, users and research institutes, in order to enable information exchange among heterogeneous systems. It was designed to support a wide range of systems, ranging from sensor device to enterprise server. The services of OPC UA ensure a seamless flow of information among multiple heterogeneous entities of an industrial automation system. OPC UA performs this seamless exchange of information by a typical client-server model, and with a platform agnostic approach. Information is transacted between OPC UA client and server.

OPC UA standard is developed and maintained by the OPC foundation [20]. Its data modelling and object-oriented techniques allow the modelling of any kind of information data. The specifications of the OPC UA standard [4] provide the autonomy for defining industry or standards organization, a specific information model for exchanging information across various kinds of platforms.

## IV. Proposed Architecture

Our proposed architecture is presented in Fig. 2. With the help of OPC UA technology, the industrial automation system is enabled to establish connectivity with an IoT infrastructure.

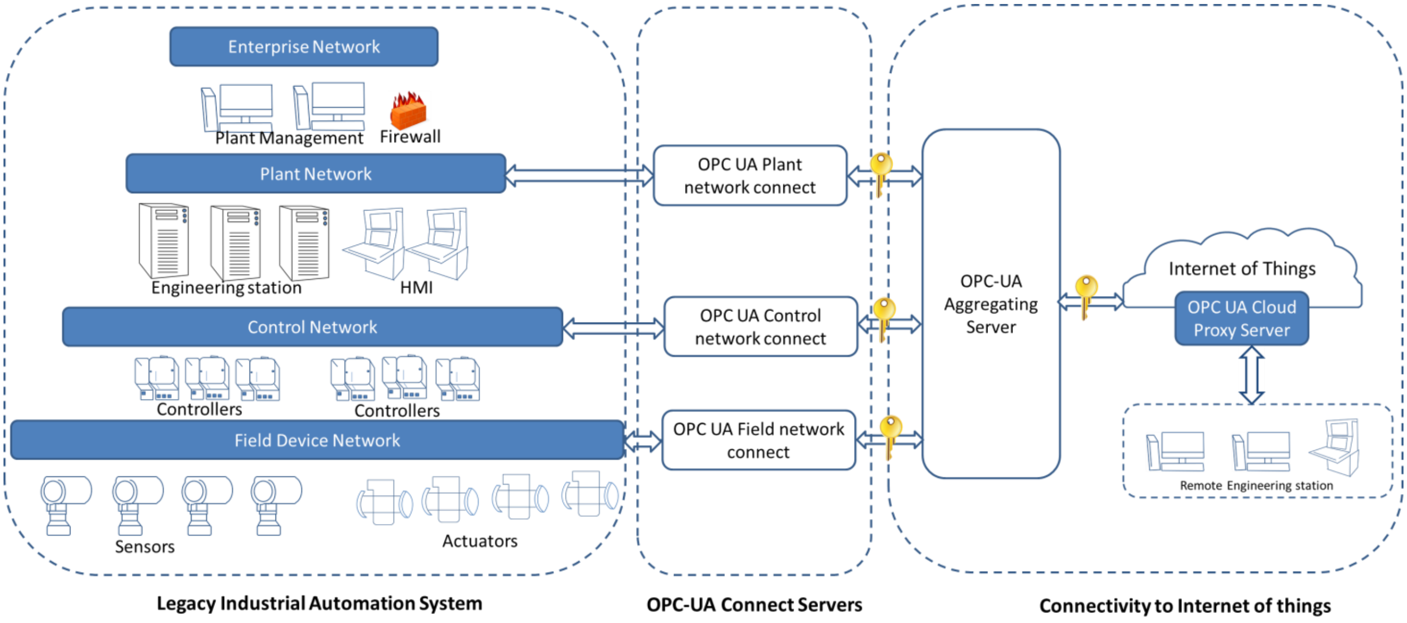

Enterprise Network
Plant Management
Firewall
Plant Network
Engineering station
HMI
Control Network
Controllers
Controllers
Field Device Network
Sensors
Actuators
OPC UA Plant network connect
OPC UA Control network connect
OPC UA Field network connect
OPC-UA Aggregating Server
Internet of Things
OPC UA Cloud Proxy Server
Remote Engineering station
Legacy Industrial Automation System
OPC-UA Connect Servers
Connectivity to Internet of things

Figure 2. Industrial Automation connectivity with an IoT infrastructure.

OPC UA enables, not only pure process data exchange, but also semantic information exchange as well. This architecture is built around three main modules as shown in Fig. 2, i.e. OPC UA Connect server, OPC UA Aggregating server and OPC UA Cloud proxy Server. Three different OPC UA Connect servers, i.e. OPC UA Plant network connect, OPC UA Control network connect and OPC UA Field network connect, are used for modeling the data at the Plant network layer, Control network layer, and Field Device network layer respectively. These three OPC UA Connect servers are leveraged to model the data from the devices located at the factory floor to enterprise applications. The OPC UA aggregating server aggregates the data from the OPC UA Connect servers into one place. It is helpful for any OPC UA client application to access the data from multiple OPC UA servers from a single node. This feature is thus facilitated by the OPC UA aggregating server. OPC UA aggregating server also provides a mechanism for chaining the data from multiple OPC UA servers to a particular OPC UA client application, for limiting functions and data accessibility. OPC UA cloud proxy server provides the proxy connection between OPC UA aggregating server and remotely located different OPC UA client applications such as Enterprise Management System, remote engineering workplace, etc. By divulging the different layers of industrial information data to an IoT infrastructure, an industrial automation system will be capable of exchanging information with other applications such as ERP and MES. Thus, with this architecture, an industrial automation system can be enhanced with a capability for handing dynamic business processes. For example, if a supplier delays the supply of raw materials, with an IoT enabled infrastructure, this information can be conveyed to a production site as quickly as possible. Instead of shutting down the production, the production speed can be slowed down and other maintenance activities can be performed in parallel.

## V. OPC UA Security Model Analysis

Security model of OPC UA is a very important aspect for its reliable operation in an industrial automation plant. A security compromised OPC UA server and client application can result into devastating effects on the plant, causing huge financial damages, and could even lead to severe health hazards. The security requirements for OPC UA has been addressed in its standard. This section provides a short overview of the existing security specifications of OPC UA from its reference security standard [21]. OPC UA features the basic security primitives i.e. authentication, authorization, integrity, confidentiality and auditability of data as illustrated below.

### A. *Authentication*

In OPC UA, both the server and client application establish a mutual trust between them by validating each other’s identities. This verification and validation of each other’s identity is performed on the basis of X.509 based certificates [22]. Thus, when an OPC UA client application wants to establish a connection with the OPC UA server side application, the server application performs an authentication check of the client application, and vice versa on the basis of X.509 based certificates. Thus, a X.509 certificate acts like an identity of the OPC UA application.

### B. *User authentication*

OPC UA server guards the user access control by validating each user’s identity. This can be done with the help of username & password or by a X.509 based certificate based token. OPC UA server checks the authenticity of the users to permit only legitimate users to access the server ontents. When an end user tries to access the OPC UA server from an OPC UA client application through a network, the OPC UA client securely sends the credentials of the user like his username and password to the OPC UA server. The communication traffic between OPC UA server and OPC UA client is made secure by Transport Layer Security (TLS) [23] protocol.

### C. *User authorization*

OPC UA also features authorization controls of users. User can perform only those job functions as per their set privileges.

### D. *Secure communication*

OPC UA client and OPC UA server communicate over a secure channel. The messages that are exchanged between OPC UA server and OPC UA client are digitally signed to provide authentication and integrity. Messages are optionally encrypted to provide confidentiality as per need basis. TLS protocol [23] is used for providing these security features to the exchanged messages.

### E. *Auditability*

Event logging is supported in OPC UA for recording important user and system activities such as user access logs and communication logs on the OPC UA server, when OPC UA clients connects to it.

### F. *Security policy management*

Security policies like cryptographic key sizes, type of crypto algorithms, key expiry time etc. is maintained in Cyber Security Management System (CSMS) [21]. Thus, the security policies of the OPC UA server are managed centrally from this management system. When an OPC UA client tries to connect to a OPC UA server, it performs a discovery mode by sending discovery messages to obtain the security profiles that are supported by the OPC UA server. This helps the OPC UA client to detect, and accordingly use the security policies of the OPC UA server for establishing a secure connection to it.

### G. *Availability*

OPC UA server incorporates protection against message flooding, by limiting the processing of concurrent messages. Security against replay attacks is done by specifying sequence numbers and time stamping for each transacted messages.

## VI. OPC UA Security Model Analysis From An IoT Prospective

When using OPC UA in the IoT computing space, the probability of a security attack towards the OPC UA based systems will be larger. This is because the world of Internet is exposed to different types of sophisticated cyber threats. Additionally, the number of cyber security threats will grow, and propagation of new threats can also occur from the Internet domain. Hence, it is important to increase the security level of the OPC UA to defend against the Internet based attacks. In this section, we analyze the existing security workflow and specifications of OPC UA, and propose some ideas for enhancing the security of OPC UA, from an IoT perspective. Following are the security enhancement areas which we propose.

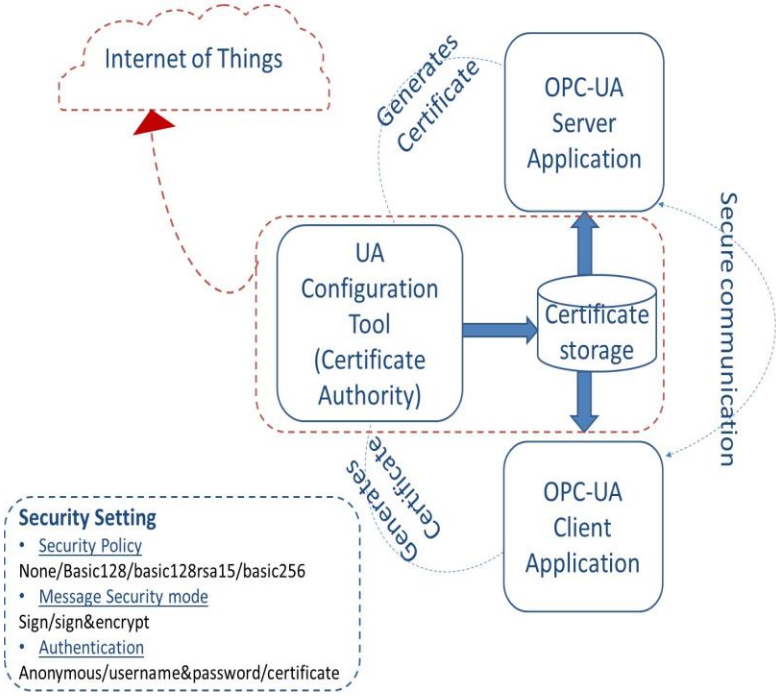


Figure 3. Certificate management of OPC UA.

### A. *Certificate management in IoT cloud*

X.509 based digital certificates are recommended to attest the identity of OPC UA applications, as per its standard prescribed security specifications [21]. During communication, an OPC UA client and server application shall thus perform an authentication check on each other, with the help of their X.509 based digital certificates.

In Fig. 3, we describe the X.509 certificate generation and issuing process for the OPC UA server and client applications. Presently, the existing OPC UA configuration tool has a built in Certification Authority (CA), which generates certificates for the OPC UA applications, and stores them in a certificate storage Database.

During the course of certificate generation process, the administrator of OPC UA configuration tool registers the credentials of each OPC UA application, and subsequently also generates and stores their issued certificates in a certificate storage database. The credentials of OPC UA applications could be the application name, organization that has built the application, etc. Then, each OPC UA server and client application accesses this database to obtain their corresponding issued certificates, and further uses their certificates to securely communicate among each other. In the present state, the CA and certificate database of an OPC UA based system are managed and operated locally within the perimeter of an industrial automation plant.

As we intend to integrate the existing industrial automation system with an IoT infrastructure by using the complementary OPC UA technology, we propose to shift and deploy the CA functionality and certificate storage database of the existing OPC UA based systems into the IoT's cloud computing space. This proposal of shifting is depicted using the red dotted mark, as referred in Fig. 3. The benefit of shifting the OPC UA certificate management functionalities into the IoT cloud is that it will enable to manage the ability to manage the certificates of various OPC UA server and client applications from any geographical location. This will increase the flexibility and scalability of managing the certificates of OPC UA applications.

However, operating the certificate management feature in the IoT cloud space has some business challenges too. Typically, each industrial automation plant tries maintaining and operating its CA in its own corporate network. This is done in order to maintain the certificate related security policies within its perimeter. Therefore, the existing industrial automation plant owners may not be ready and flexible enough to move their CAs into the IoT cloud space.

### *B. Secure registration of OPC UA applications*

We also suggest to consider a secure registration of OPC UA application credentials in the CA, which is managing the certificates of OPC UA applications. As explained before, that in the existing scenario, the OPC UA configuration tool registers the credentials of each OPC UA application, and subsequently generates certificates for the registered applications with its inbuilt CA. Before registration, neither the OPC UA configuration tool nor the administrator of the tool, verifies whether or not the credentials of OPC UA applications are valid. Checking the authenticity of these credentials is crucial for the initial trust establishment of legitimate OPC UA applications with the CA. Before registering each credential of OPC UA applications in the CA, it is important to check their authenticity. If the registered credentials belong to a rogue OPC UA application, the malicious application will manage to legitimately obtain a certificate from the CA. An attacker can further use the rogue application to inject attacks and perform harmful operations on the industrial automation system. The legitimate credentials of OPC UA applications should also be stored and accessed securely. Access to the legitimate credentials of OPC UA applications should be granted only to authorize people. If these credentials reach the hands of an attacker, then the attacker can register a rogue application with the same legitimate credentials in the CA, which is managing the certificates of OPC UA applications.

### *C. Secure management of time synchronization messages*

If a time server from the Internet domain is used for synchronizing the timings of OPC UA devices, it is also important to validate the authenticity of time server and secure the time synchronization messages. A rogue time server can synchronize the OPC UA device with incorrect timings, which could negatively affect certain automation applications. These could have devastating effect on automation applications, whose operations are dependent on accurate timings. Denial of Service (DoS) attack can also be injected in the OPC UA enabled devices by improper expiration of OPC UA application certificates. This can be caused by wrong device timings. If the certificate of a OPC UA server application expires before its lifetime, then OPC UA clients may face issues in connecting to the OPC UA server due to certificate expiry error.

### *D. Enhance user authentication and authorization*

Introducing OPC UA in the IoT space will certainly also increase the number of legitimate users, who can access the OPC UA server and client applications from the Internet. A secure management of such increasing number of user credentials is important for allowing only legitimate user operations on the OPC UA.

### *E. Secure web services*

OPC UA web servers should also adopt security measures against Cross-site Scripting (XSS) and code injection attacks [24]. In such attacks, the attacker tries injecting malicious scripts in a legitimate web server. By injecting such rogue scripts, the legitimate OPC UA web server can malfunction or be directed to execute devastating operations. Such rogue operations include illegitimately changing the configuration parameters of the OPC UA enabled industrial automation device. This can make the device perform unintended and rogue operations inside the automation plant. XSS and code injection based attacks are becoming very popular web attacks in the Internet space [25]. Therefore, OPC UA web server should also consider and implement countermeasures against such sophisticated web attacks.

Proper sanitization of data within the OPC UA web servers is also recommended for securing against such Internet based web attacks. Also, disabling the unnecessary web services of OPC UA is a good security practice to reduce the attacking surface for the Internet attackers.

### *F. Categorization and securing event loggings*

Using OPC UA in the IoT space would create a mixture of OPC UA type events and Internet based events, like the traditional IT related events. For ensuring a well-structured audit log, a clear separation for these two types of event would make the event logs more readily understandable to the operators of the industrial plant. For example, a OPC UA system monitoring engineer would be interested to view the OPC UA event logs instead of IT type events.

Categorization of event logs would enable the OPC UA engineers to quickly view the OPC UA event type data. It is also required to securely transport the event log messages to the OPC UA event logging server. This can be done by digitally signing and optionally encrypting them, when required. This is important especially if the OPC UA event logging and management server is operating in the IoT cloud. It will prevent the attackers from the Internet to produce fake or to modify the legitimate event log messages, which could create a wrong status of the industrial automation system.

## VII. Conclusion

IoT is a new technology revamp, which will provide the infrastructure for exchanging the information across different entities. For an industrial automation system, IoT is the one key enabler for facilitating dynamic business and engineering processes. We have analyzed and proposed an architecture which describes how the industry proven OPC UA technology can be used for evolving, and thus enabling an industrial automation system to exchange information with an IoT infrastructure. We have also analyzed the existing security model of OPC UA from an IoT's perspective. Based on our security analysis, we have proposed and thus highlighted the suggested areas where security improvements are required when using OPC UA in the IoT's space. The performance verification and validation of our proposed architecture needs to be thoroughly tested in a real industrial automation system. Our future research work shall focus towards such experimental evaluations of our proposed architecture.

## References

[1] Galloway, Hancke B, "Introduction to Industrial Control Networks," Communications Surveys & Tutorials, IEEE , vol.15, no.2, pp.860,880, Second Quarter 2013

[2] http://www.erp.com/component/content/article/324-erp-archive/4407-erp.html [accessed May 2015]

[3] McClellan, Michael (1997). Applying Manufacturing Execution Systems. Boca Raton, Fl: St. Lucie/APICS. ISBN 1574441353.

[4] OPC UA Specification, Part-1 Overview and Concepts. Reterived from https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/ [accessed Feb 2015]

[5] Kevin A (22 June 2009). That Internet of Things Thing, in the real world things matter more than idea. Reterived from http://www.rfid-.journal.com/articles/view?4986 [accessed December 2014]

[6] Xu Li,He Wu,Shancang Li, "Internet of Things in Industries: A Survey," Industrial Informatics, IEEE Transactions on , vol.10, no.4, pp.2233,2243, Nov. 2014 doi: 10.1109/TII.2014.2300753

[7] Kim M., Hwang J, "New approach for Convergence of IT + pharmaceutical industry," Information and Communication Technology Convergence, (ICTC), 2010 International Conference on , pp.569,570, 17-19 Nov. 2010

[8] Song Bo, Xing Qian, "On security detecting architechture of food industry based on Internet of Things," Automation and Logistics (ICAL), 2011, IEEE International Conference on , pp.81,85, 15-16 Aug. 2011

[9] Tan L,Wang N, "Future internet: The Internet of Things," Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on , pp.V5-376,V5-380, 20-22 Aug. 2010

[10] Chen Y, "Challenges and opportunities of internet of things," Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific, pp.383, 388, Jan. 30 2012-Feb. 2 2012

[11] Imtiaz, J., Jasperneite, J., "Scalability of OPC-UA down to the chip level enables "Internet of Things"," Industrial Informatics (INDIN), 2013 11th IEEE International Conference on , pp.500,505, 29-31 July 2013

[12] Copie A, Fortis T, Munteanu, V.I, "Benchmarking cloud databases for the requirements of the Internet of Things," Information Technology Interfaces (ITI), Proceedings of the ITI 2013 35th International Conference on , pp.77,82, 24-27 June 2013

[13] Recommendations for implementing the strategic initiative INDUSTRIE 4.0, Acatech, April 2013. Reterived from http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf

[14] Perera C., Liu C.H., Jayawardena S., Min Chen, "A Survey on Internet of Things From Industrial Market Perspective," *Access, IEEE*, pp.1660,1679, 2014 doi: 0.1109/ACCESS.2015.2389854

[15] Singh D., "Developing an architecture: Scalability, mobility, control, and isolation on future internet services," Advances in Computing, Communications and Informatics (ICACCI), 2013

[16] Ungurean I., Gaitan N., Gaitan V G., "An IoT architecture for things from industrial environment," Communications (COMM), 2014 10th International Conference on , pp.1,4, 29-31 May 2014

[17] http://www.opcconnect.com/dotnet.php [accessed Jan 2015]

[18] Keoh S, "Securing the Internet of Things: A Standardization Perspective", IEEE Internet of Things Journal, Vol. 1, No. 3, June 2014

[19] Sajjad S M, Yousafy M, "Security Analysis of IEEE 802.15.4 MAC in the context of Internet of Things (IoT)", Proceedings of the 2014 Conference on Information Assurance and Cyber Security (CIACS)

[20] https://opcfoundation.org/ [accessed Jan 2015]

[21] OPC UA Security Standard Specifications, OPC UA Specification Part 2: Security Model Release. Reterived from https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-2-security-model [accessed Feb 2015]

[22] RFC 2459, X.509 Certificate.Reterived from https://www.ietf.org/rfc/rfc2459 [accessed Feb 2015]

[23] RFC 5246, Transport Layer Security (TLS) Protocol, Version 1.2. Reterived from https://datatracker.ietf.org/doc/rfc5246 [accessed Feb 2015]

[24] Cross-site Scripting, OWASP guide to XSS attacks. Reterived from https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29 [accessed Feb 2015]

[25] Web Hacking Incident Database (WHID) 2013 statistical records for XSS attacks. Reterived from http://projects.webappsec.org/w/page/13246995/Web-Hacking-Incident-Database#RealTimeStatistics [accessed Mar 2015]

# Peer to Peer Media Management for Augmented Reality

Raimund K. Ege

Dept. of Computer Science
Northern Illinois University
DeKalb, IL, USA
ege@niu.edu

***Abstract*—Immersion into rich multimedia is now the norm in today's Internet. Combining multiple streams of textual, audio and media data from a variety of sensors and sources, allow the presentation of a world that is almost "real". Users equipped with portable and wearable devices can become consumers and contributors to an augmented reality. In this paper we describe a general software architecture for an augmented reality immersion network based on crowd sourced media gathering and distribution. We describe a prototypical implementation with cloud and mobile components to establish a secure sharing network, to coordinate and to synchronize the media streams. Joining the content sharing network is subject to peer-to-peer trust management to protect the content and the participants.**

***Keywords-Android; augmented reality; multi-media content delivery; securing trust; peer-to-peer systems;***

## I. Introduction

What was a personal digital assistant became a smartphone and now an ever-present wearable device: with computing power, display and recording capabilities, and – foremost – with broadband connectivity. The days of just calling and texting on a smart phone are gone: we now watch TV shows, check the world-wide-web, or play games. We can participate in a host of social applications that are rich in multimedia exchange.

Modern mobile devices feature multiple input sensors, like cameras, and advanced geo-location positioning systems that us GPS, cell tower triangulation, compass and accelerometers. Users of such mobile devices are not limited to media consumption, but are allowed to become an active player in the production and sharing of media. The computing power and network connectivity enable the provision of peer-to-peer (P2P) content delivery networks: rather than just down- or up-loading media to one site, media can be shared in such P2P network at a much higher throughput, i.e. no single source bottleneck, and without central control, i.e. big brother registration. The aim of our research is to allow the forming of very large P2P content sharing networks, without central control, but with provisions that instill a degree of trust into the participants.

In this paper we describe architecture for an augmented reality immersion network based on crowd sourced media gathering and distribution. We describe an application framework with cloud and mobile components to establish a secure sharing network, to coordinate and to synchronize the media streams. Joining the peer-to-peer (P2P) content sharing network is subject to trust management to protect the content and the participants.

Possible application scenarios for this technology span from massively online multi-player games to first responder support gear for emergency personnel. In a multi-player game scenario, participants can wander a partially populated game room, that is further augmented with virtually-real objects and events that need to be handled by groups of game players. Each player carries a smart phone which transmits its sensed data (video, location) to others, and receives a coordinated and merged augmented reality view of the game room. In a first responder scenario, support gear collects and transmits sensor data to members of the response team, and receives a coordinated and merged augmented reality of the emergency scenario.

Section 2 gives some background on smartphone technology, P2P content delivery and sharing networks, and security issues such as access control, identity and trust management. We also relate our work to current research. Section 3 discusses how to manage and secure shared content, specifically which elements of security to ensure confidentiality, integrity and availability are available to mobile platforms, with a specific focus on what is available to Android smartphones. Section 4 elaborates on our P2P content sharing model, especially on how our approach defines and gauges trust, and how such trust is maintained, secured and shared in a central-server-less P2P environment. Section 5 outlines our prototype implementation with Java peers, including peers running on Android smartphones. The paper concludes with some lessons we learned and our future perspective.

## II. Background

Personal digital assistants have come a long way in recent years. The current crop of smartphones is just a stepping stone in the advance of digital devices that enable individuals to compute and connect. Wearable connected computing devices, such as wristwatches and even eye glasses (Google GLASS) are available. The focus is shifting from computing and storage capabilities on these devices to connectivity and multi-media input and output components.

Connectivity capabilities are typically wireless and include high-bandwidth cellular (4G, LTE) and WLAN (IEEE 802.11) connections, plus lower-bandwidth near field connections (Bluetooth, NFC, etc.). Transmission rates in the multi megabits per second range and latency rates in the sub millisecond range are currently quite standard. Multi-media I/O components include high-definition screens and video cameras, high-fidelity speakers and microphones. Plus components to determine device location, position, and attitude: GPS, accelerometers, compass, etc.

Consider the Google Maps application on a smartphone: the smartphone acquires its location via GPS, sends the location to a Google server. The server responds with appropriate map data which is displayed on the phone. As the user moves, the map data is updated. Such a simple example of augmented reality can be further improved by adding real-time traffic data from traffic sensors. Moreover, data gathered from other smartphones can augment the display with a multitude of other useful data, as in the Waze (www.waze.com) navigation application. Augmenting map data with imagery from satellites and street based cameras (Google Street View) is already common practice. Adding video and other sensor data from nearby smartphones is the logical next step.

Augmented reality provides a live view of a physical, real-world environment. It can be direct or indirect. Its elements are supplemented or augmented by computer-generated input from sensors such as sound, video, graphics or location data. While this field of research has quite a long history [1], only recently has the computing and bandwidth capabilities enabled truly wide acceptance [2] [3]. Key elements of such crowd-sourced augmented reality are real-time coordination of sensor data and establishment of authenticity and trust in the participating peers. Coordination of the data is achieved via ever precise location information, coupled with attitude references. Current locating sensor and accelerometer technology has shrunk and is available in state of the art smartphones.

Access control, trust and digital rights management is essential. Access control is common place in many applications. A server maintains a database of user and account information. A user gains access to the system by providing a user id with additional security information, typically a password. Once authenticated, the user is “trusted”, i.e. is allowed to participate in the system’s mission. The information stored by the server can include the users past history of participation, which in turn can be used to augment the level of trust in the user. Other users might contribute to the trust evaluation by submitting feedback on others. The level of trust might determine the level of participation a user is allowed, e.g. users with a low level of trust might be able to consume content, while users with a high level of trust might be able to contribute media.

Many modern systems outsource their central access control to an external provider. In a centralized system central access control makes sense: OpenID [4] is an example. OpenID providers maintain identity information and allow users to choose which and when to associate information with their OpenID that can be shared with sites they visit upon request. With OpenID, password information is passed to the identity provider which verifies and then confirms the identity of a user.

Peer-to-peer systems lack a central authority: peers need to collaborate and obtain services within an ad hoc environment where little trust exists. All peers collectively have to manage the risks involved in a collaboration: incomplete knowledge and little prior experience is the norm. Typical approaches address this uncertainty by developing and establishing trust among peers. Trusted third party systems [5] or self-regulating systems with community-based feedback [6] are ways to build trust.

In today's collaborative and complex world, a peer can both protect itself and at the same time benefit only if it can adjust and react to new peers dynamically and enforce access control via flexible and proper privileges. Management of trust helps minimize risk and ensures the network activity of benign entities in distributed systems [7].

Many secure content delivery systems focus on digital rights management, especially for peer-to-peer and mobile systems. Several schemes have been introduced: OMA DRM [8] – promulgated by the Open Mobile Alliance industry consortium – attempts to standardize a framework to secure media for mobile devices. Public key infrastructure (PKI [9]) style certificates are employed that contain and authenticate public keys to protect media. While we also use PKI, our intention is to go further in that we do not want to require absolute certainty of access right, but rather allow building of graduated trust which enables graduated access control to digital media.

In our prior work we focused on how trust can be quantified [10], and how trust can be managed securely [11] by peers who participate in a P2P content sharing network. In this paper we combine these approaches and add the dimension of combining multiple peers’ perspectives into one augmented reality.

## III. SECURING AND MANAGING SHARED CONTENT

The key to successful sharing of content is its security. While sharing implies to let others consume content, it has to be done in a safe and secure environment. The conventional CIA triad, i.e. confidentiality, integrity and availability, also applies in the mobile content sharing context. Shared media must not be consumed by un-authorized peers, it must stay confidential to only authorized peers. Shared media should not be altered, its integrity must be preserved. And the media, plus the data needed to make access decisions must be available to authorized and trusted peers.

Means to ensure confidentiality include encryption algorithms and protocols which are readily available on the Android platform. Android Smartphones are used daily in ecommerce apps and applications which necessitates support for all common security standards. The underlying Java system provides a rich provider architecture [12] to enable key management, key exchange, symmetric and asymmetric encryption, block and stream ciphers. Android customizes the implementation of the Java architecture via the "Bouncy Castle" [13] implementation. And of course, computing power is amply available on today's multicore smartphone systems.

We also draw on the standard Public Key Infrastructure [PKI] standard. Public and private key pairs are generated for each peer. Public keys are shared, i.e. made available to all peers that participate in the content sharing network. And, of course, private keys are maintained in secret, which enable peers to decrypt and authenticate communications.

## IV. P2P CONTENT SHARING MODEL

Peer-to-peer (P2P) is a communications model in which peers communicate on an equal basis with each other. There is no central sever, no peer is a mere client. All peers have the same capabilities: any peer can initiate a communication session.

While all peers share advanced connection capabilities with high throughput and low latency, in our architecture each peer can have all or some of the following capabilities:

1. The peer has video and audio reproduction device, i.e. a suitably-sized display screen and audio speakers. A peer that has this capability is called a “consumer” peer.
2. The peer has several sensors, such as a video camera, audio microphone, location sensors, such as GPS receiver, and attitude indicator, such as an accelerometer. A peer that has this capability is called a “producer” peer.
3. The peer has computing power to merge streams of multi-media, such as combining video, audio, and location data; but also the ability to coordinate multiple video/audio streams together based on precise location and attitude data. A peer that has this capability is called a “mediator” peer.
4. The peer has administrative authority. It can gather and keep information about the available peers, their capabilities and their trust worthiness. A peer that has this capability is called a “tracker” peer.

Each peer also carries a unique identity, which is made known to other peers.

Once a peer is identified, it is a matter of trust whether and to what degree the peer is allowed to partake in the shared media content. The trust value and the peer’s history of relevant transactions are maintained in a container we call “trust nugget”. This nugget contains detailed information on a peer’s participation, such as length and quality of stream transmission, ratio of seed vs. leech behavior, judgments of other stream participants, etc. The nugget content is signed with a special master private key. It can be verified only via the special master public key. This ensures that the trust information maintains its integrity, even as it is shared with peers in the swarm that have lower trust values.

Trust information per peer is maintained by “tracker” peers. The sole requirement for starting a new swarm is the existence of an initial tracker peer that we call the “boot strap peer”. This peer initially creates the master public/private key pair that is only shared with other trusted tracker peers. A trusted peer maintains a database of trust nuggets for all peers in the swarm. Again, initially, only one peer, i.e. the boot strap peer, has such a database, but as other peers attain higher trusted peer status, they can become tracker peers and receive the database. All tracker peers also participate in synchronizing the database to reflect the trust state of the complete P2P network and all its peers. The trust value for a peer is computed from the peer’s history of transactions. The computation is done by a tracker peer whenever a peer reports on another peer. A common scenario is that a peer serves as a producer of media content: it makes the content available to the peers in the swarm. Once a peer has “consumed” the content, the “producer” peer notifies a tracker peer of the peer’s behavior: good or bad. The tracker peer enters a new transaction into the peer’s nugget and signs it with the master private key. Tracker peers are the backbone of our trust model. New peers need to register with one trusted peer which creates a trust nugget for the new peer. The new peer also creates a public/private key pair and submits its public key to the tracker peer.

When a peer acts as a producer peer, i.e. it makes new content available to the swarm; it can set a trust threshold, i.e. a minimum trust value, required for any peer to access the content. Only peers whose trust value meets the threshold can participate. The producer peer also determines the weight of a peer’s participation when computing a peer’s new trust value.

Mediator peers transform streams of media from producer peers into new streams. In effect, a mediator peer combines “consumer” and “producer” behavior. Like any peer, it has to register with a tracker and establish a trust nugget. To “consume” a stream from a producer it must pass the trust threshold, and in turn it will set a trust threshold for other peers to consume its output stream.

Consider the following scenario to illustrate how our model enables shared augmented reality: a user holds a smartphone with forward facing camera, video display, and geolocation sensors. Here the user’s smart phone serves as a producer peer serving a video stream, a location data stream and an attitude data stream. Somewhere else in the cloud is the producer peer serving the virtual reality model of the user’s surroundings. Somewhere else in the cloud is a mediator peer that coordinates and combines the real-time video from the user’s camera with a virtual reality model of the user’s surroundings based on the location and attitude data stream from the user. And finally, the user’s smartphone also is a consumer peer in that its display shows a video stream produced by the mediator peer in the cloud, We envision content sharing networks with multiple consumer peers, mediator peers, producer peers, all coordinated by tracker peers. A peer can impersonate one tracker personality, e.g. just be a plain consumer peer, but also serve as the all might peer in combining all four peer personality. And of course, a peer can add and shed personalities as the situation and context changes.

## V. IMPLEMENTATION FRAMEWORK

In reflection of our content sharing network architecture, our implementation framework provides feature rich components that can be assembled into a peer. At first, a peer has the basic capabilities to establish its identity, its peer properties and to connect to other peers. Then each peer can assume additional capabilities via any of these components:

1. The "producer" component, which gathers data from sensors (i.e. camera, microphone, GPS receiver, accelerometer, etc.) and make them available in stream format.
2. The "mediator" component, which receives data on multiple incoming streams, to produce a combined outgoing stream, which contains the logical coordination of the incoming data. The coordination is based on location and attitude data that is associated with input streams. The coordination can be done in several modes: add, merge, and layered. The add mode simply combines that input streams: this is useful for time synchronized multimedia that does not cause direct interference, e.g. combining audio and video. The merge mode attempts to combine similar-type input streams into an out stream. Time and location data is used, plus an attempt is made to recognize key features that are present in all input streams to correct the location and attitude data. The layer mode preserves the input streams and allows consumers of the output stream to select layers dynamically.
3. The "consumer" component, which receives an incoming stream and renders it onto suitable output devices (i.e. display, speaker, etc.). If the input stream is of layered mode, it also allows selecting one or more layers.
4. The "tracker" component, which accepts registrations from other peers, maintains their trust information and coordinates the available peers and streams available in the content sharing network.

All these components are available in Java, so they can be assembled into a peer that runs on a mobile device, i.e. Android smartphone, or a peer that resides in the cloud.

For our prototype implementation we assembled a set of peer types built from these components:

(1) A set of tracker peers, initially just one: the boot strap peer application; this application runs as a Java application in the cloud and also serves as the control and observation point for our prototype implementation.

(2) A producer peer Java application to submit information about a content stream; multiple instances, i.e. multiple source peers can be introduced into the content sharing network.

(3) A mediator peer Java application that coordinates multiple data streams, and makes it available to other peers.

(4) A consumer peer to run on an Android mobile device. Android is implemented in Java on a Linux base and therefore offers a flexible and standard set of communication and security features.

Figure 1 shows a sample scenario with one producer, one tracker, one mediator, and one consumer peer:

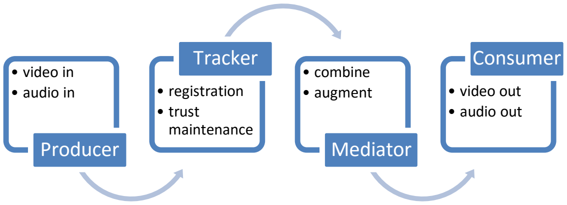


Figure 1: Peer-to-Peer Network

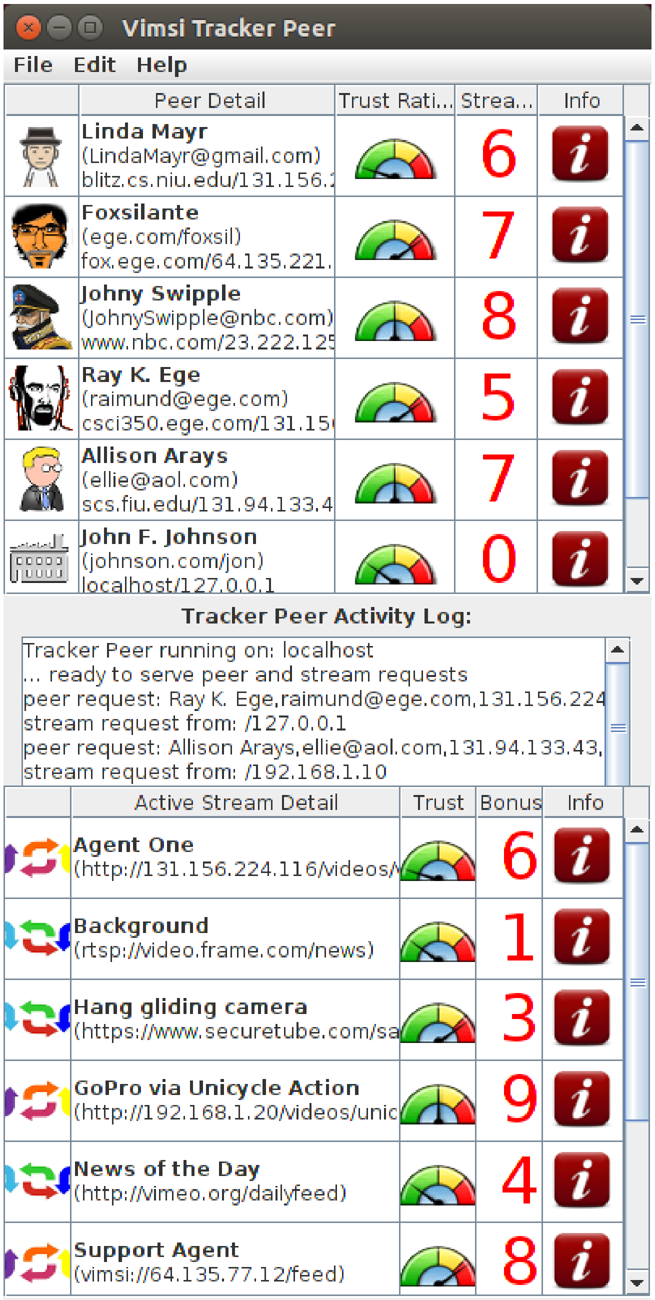
### A. *Tracker Peer (also boot strap peer)*

The central component of our architecture is the tracker peer. It maintains a database of all peers and a tracks the collection of data streams that are made available by sources. Our tracker peer prototype presents a display of all peers and streams (see Figure 2).

When a new peer connects to a tracker peer, authentication is achieved via the peer's openID, which is validated the openID provider. If the peer is new, i.e. the tracker peer has no trust nugget for the peer, the new peer must provide its public key and a new trust nugget is created. The peer's public key is later provided to consumer peers who will use it to encrypt content destined for that peer. The top part of the tracker window shown in Figure 1 lists the peers that are currently part of the content sharing network. Each peer is shown with its avatar, its identification and location detail, the level of trust it has achieved so far, and the number streams that the peer is currently participating in. The center part of the tracker window shows a log of peer and stream access activity among the peer that are part of the content sharing network. The lower part of the tracker window shows the list of available media streams. The active stream detail column shows the title and the actual URL used to connect to the stream. The "Trust" column displays the minimum trust threshold that a peer must pass to be allowed to participate in the stream. The "bonus" column list the increment a peer is giving for a successful, i.e. benevolent, participation in the stream production, delivery, and consumption.

### B. *Producer Peer*

The producer peer application is used to submit information about a content stream; multiple instances, i.e. multiple producer peers can be introduced into the content sharing network. In our prototype implementation, we provide a very simple version: a simple dialog that captures a input sensor as media stream and allows to submit the stream information to a trusted peer. Figure 3 shows a simple Java application as producer peer.



| | Peer Detail | Trust Rati... | Strea... | Info |
|---|---|---|---|---|
| | **Linda Mayr**<br>(LindaMayr@gmail.com)<br>blitz.cs.niu.edu/131.156.1 | | 6 | i |
| | **Foxsilante**<br>(ege.com/foxsil)<br>fox.ege.com/64.135.221. | | 7 | i |
| | **Johny Swipple**<br>(JohnySwipple@nbc.com)<br>www.nbc.com/23.222.125 | | 8 | i |
| | **Ray K. Ege**<br>(raimund@ege.com)<br>csci350.ege.com/131.15 | | 5 | i |
| | **Allison Arays**<br>(ellie@aol.com)<br>scs.fiu.edu/131.94.133.4 | | 7 | i |
| | **John F. Johnson**<br>(johnson.com/jon)<br>localhost/127.0.0.1 | | 0 | i |

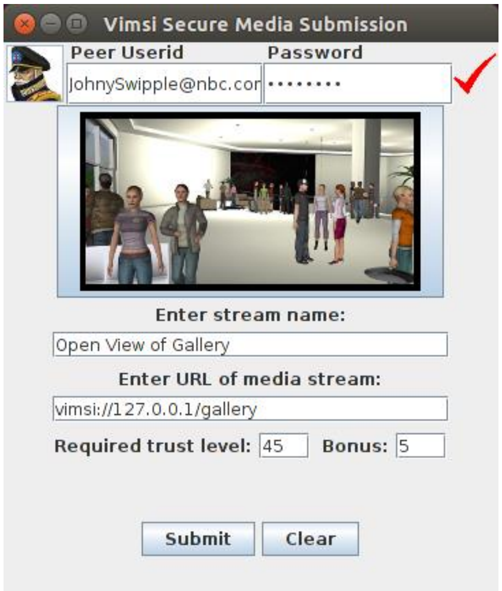
```
Tracker Peer running on: localhost
... ready to serve peer and stream requests
peer request: Ray K. Ege,raimund@ege.com,131.156.224
stream request from: /127.0.0.1
peer request: Allison Arays,ellie@aol.com,131.94.133.43,
stream request from: /192.168.1.10
```

| | Active Stream Detail | Trust | Bonus | Info |
|---|---|---|---|---|
| | **Agent One**<br>(http://131.156.224.116/videos/ | | 6 | i |
| | **Background**<br>(rtsp://video.frame.com/news) | | 1 | i |
| | **Hang gliding camera**<br>(https://www.securetube.com/sa | | 3 | i |
| | **GoPro via Unicycle Action**<br>(http://192.168.1.20/videos/unic | | 9 | i |
| | **News of the Day**<br>(http://vimeo.org/dailyfeed) | | 4 | i |
| | **Support Agent**<br>(vimsi://64.135.77.12/feed) | | 8 | i |

Figure 2: Boot Strap Peer

### C. *Mediator Peer*

The purpose of a mediator peer is to select input streams and coordinate them into an output stream. Figure 4 shows a screen capture of the Java Mediator Peer prototype. Once the peer is authenticated with a tracker peer, it requests a list of available streams. Figure 4 shows all streams that are currently available. Note that some streams are not currently available: they display a “do not touch” symbol to indicate that they require a greater trust value for access. The reason why all stream are displayed, even the ones which require a higher trust value than what the peer currently has, is to give the peer an incentive to first participate in another stream to add the bonus to its trust value. However, only streams can actually be selected for which the peer is currently qualified.



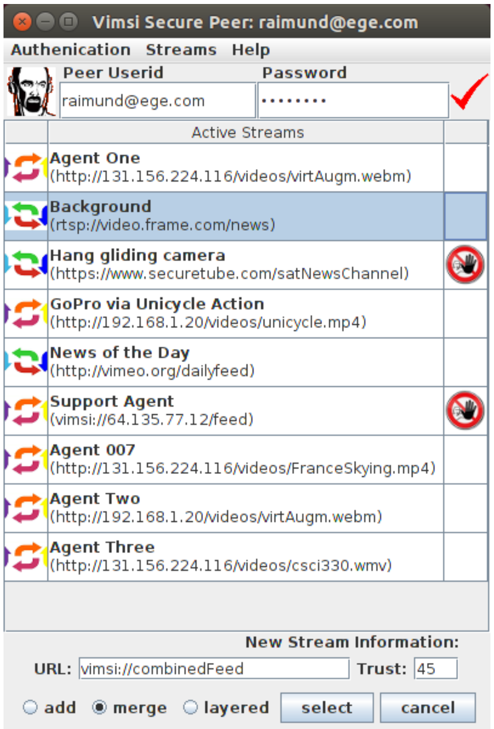
Peer Userid: JohnySwipple@nbc.cor
Password: ••••••••

Enter stream name: Open View of Gallery

Enter URL of media stream: vimsi://127.0.0.1/gallery

Required trust level: 45 Bonus: 5

Figure 3: Media Producer Peer


Vimsi Secure Peer: raimund@ege.com
Authenication Streams Help
select
cancel

Peer Userid: raimund@ege.com
Password: ••••••••

| Active Streams | |
|---|---|
| **Agent One**<br>(http://131.156.224.116/videos/virtAugm.webm) | |
| **Background**<br>(rtsp://video.frame.com/news) | |
| **Hang gliding camera**<br>(https://www.securetube.com/satNewsChannel) | ✋ |
| **GoPro via Unicycle Action**<br>(http://192.168.1.20/videos/unicycle.mp4) | |
| **News of the Day**<br>(http://vimeo.org/dailyfeed) | |
| **Support Agent**<br>(vimsi://64.135.77.12/feed) | ✋ |
| **Agent 007**<br>(http://131.156.224.116/videos/FranceSkying.mp4) | |
| **Agent Two**<br>(http://192.168.1.20/videos/virtAugm.webm) | |
| **Agent Three**<br>(http://131.156.224.116/videos/csci330.wmv) | |

New Stream Information:

URL: vimsi://combinedFeed Trust: 45

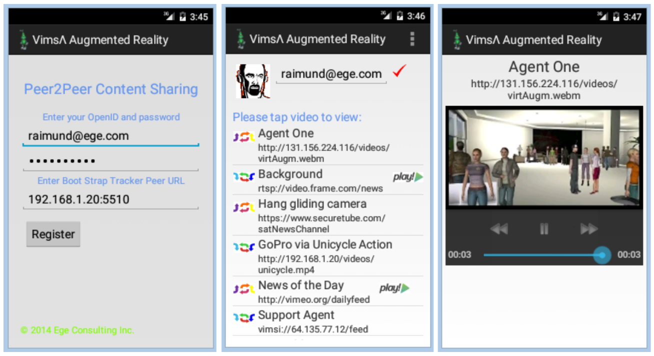
○ add ● merge ○ layered

Figure 4: Mediator Peer

The mediator peer further has to set which mode should be used for the coordination: add, merge or layered; “merge” is selected here to indicate that the selected input streams are time and location coordinated and merged into a combined output stream. Finally the dialog allows entering a unique name for the combined stream. The new stream has the tag “vimsi” which indicates to other peers that it is a combined stream coordinated by a relay peer.

### D. Consumer Peer

The final component of our prototype framework is our proof-of-concept consumer peer implementation for the Android platform. Figure 5 shows three screens: “login”, “stream selection”, and “stream play” of our Android prototype consumer peer application.



3:45
VimsΛ Augmented Reality
Peer2Peer Content Sharing
Enter your OpenID and password
raimund@ege.com
••••••••••
Enter Boot Strap Tracker Peer URL
192.168.1.20:5510
© 2014 Ege Consulting Inc.

3:46
VimsΛ Augmented Reality
raimund@ege.com ✓
Please tap video to view:
Agent One
http://131.156.224.116/videos/virtAugm.webm
Background
rtsp://video.frame.com/news
Hang gliding camera
https://www.securetube.com/satNewsChannel
GoPro via Unicycle Action
http://192.168.1.20/videos/unicycle.mp4
News of the Day
http://vimeo.org/dailyfeed
Support Agent
vimsi://64.135.77.12/feed

3:47
VimsΛ Augmented Reality
Agent One
http://131.156.224.116/videos/virtAugm.webm
00:03 00:03

Figure 5: Android Prototype Peer App

The “login” screen allows the peer to authenticate with its OpenID credentials. The user enters userid and password, plus the URL of a boot strap tracker peer. If the peer is recognized into the content delivery network, the tracker peer transmits all available streams to the new peer. The “stream selection” screen shows these streams. As before, not all streams are available to the new peer: only those that display the “play” button can be used by this peer based on its trust level. Once the “play selected video stream” button is pressed, and a sufficient read-ahead buffer has been accumulated, the video stream starts playing on the Android device. The third screen capture shows the video stream being displayed. The video shown here is derived from a scene generated by a virtual reality rendering producer peer. The on-screen control allow the user to control the video display.

## VI. Conclusion

Our goal was to enable the merging of realities - both real and virtual - into a comprehensive experience to enable life like immersion in real time. In this paper we described a framework for a peer-to-peer based content sharing network where peers collect, augment and share multi-media streams of data.

We introduced a model to gather, manage and use trust information to allow an ad hoc assembly of peers, and demonstrated the feasibility of our approach with a Java-based prototype implementation that includes a peer client for the Android platform. We also showed that the security capabilities of the Android/Java/Linux system are up to par and implementable on today's crop of smartphones.

While our current implementation already allowed the merging of multiple streams, much additional work needs to be done to allow the combination and mediation of real-time multimedia stream. Our next step will be to focus on using virtual reality models as concrete reference and marking points to enable realistic augmented reality worlds.

## References

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre. Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications (CGA) 21(6):34-47, 2001.

[2] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond and D. Schmalstieg. Real-Time Detection and Tracking for Augmented Reality on Mobile Phones, IEEE Trans. Vis. Comput. Graph., 16(3):355-368, 2010.

[3] A. Morrison, A. Mulloni, S. Lemmelä, A. Oulasvirta, G. Jacucci, P. Peltonen, D. Schmalstieg and H. Regenbrecht. Collaborative use of mobile augmented reality with paper maps, Journal on Computers & Graphics (Elsevier), 35(4):789-799, 2011.

[4] OpenID, http://www.openid.net. [accessed September 19, 2014]

[5] J Y. Atif. Building trust in E-commerce. IEEE Internet Computing, 6(1):18–24, 2002.

[6] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. Communications of the ACM, 43(12):45–48, 2000.

[7] H. Li and M. Singhal. Trust Management in Distributed Systems. Computer, vol. 40, no. 2, pp. 45-53, Feb. 2007.

[8] OMA Digital Rights Management V2.0, http://www.openmobilealliance.org/technical/release_program/drm_v2_0.aspx. [accessed September 20, 2014]

[9] [10] C. Adams and S. Lloyd. Understanding PKI: concepts, standards, and deployment considerations. Addison-Wesley Professional. ISBN 978-0-672-32391-1. 2003.

[10] Raimund K. Ege. OghmaSip: Peer-to-Peer Multimedia for Mobile Devices. The First International Conference on Mobile Services, Resources, and Users (MOBILITY 2011), pages 1-6, Barcelona, Spain, October 2011.

[11] Raimund K. Ege. Secure Trust Management for the Android Platform. International Conference on Systems (ICONS 2013), Seville, Spain, January 2013.

[12] Java Cryptography Architecture (JCA) Reference Guide. http://docs.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html. [accessed September 20, 2014]

[13] The Legion of the Bouncy Castle. http://www.bouncycastle.org/java.html. [accessed September 20, 2014]

# An Improved Control Algorithm for a Class of Photonics Timeslot Interchanger

Luai E. Hasnawi and Richard A. Thompson
Graduate Telecommunications and Networking Program,
School of Information Sciences, University of Pittsburgh,
Emails: {leh31,rthompso}@pitt.edu

***Abstract*—All-Optical Networks (AONs) are an active research area for use in Optical Transport Networks (OTNs) and data centers. Adding Optical Time Division Multiplexing (OTDM) to AONs can more efficiently utilize the enormous amount of bandwidth in the fibers. Photonic Timeslot Interchangers (PTSIs) are used in OTDM networks to interchange/switch timeslots. This paper proposes a control algorithm for a previously proposed PTSI that uses multiple feed-forward fiber delay-lines. This algorithm further reduces the required number of delay elements needed for non-blocking operation from the reduction previously reported under a less optimal algorithm. Any reduction in the number of PTSI components should result in increased output signal power, reduced footprint, and reduced manufacturing cost.**

***Keywords–Photonic Switching; Optical Time Switching; Photonic Timeslot Interchanger; Circuit Switched; Control Algorithm.***

## I. Introduction

Cloud computing services are solutions to minimize the initial cost of building new businesses. The enormous amount of data that is processed, stored and shared in the cloud is beyond the capacity of common copper lines. Lately, optics researchers have been proposing a high speed switching system to connect server racks by fiber optics [1][2].

The inflation of file sizes, as well as the transmission rates required by recent applications, has attracted researchers attention to increasing network utilization. Cisco forecasts that IP traffic will reach 1.6 zettabytes per year by 2018 [3]. A common practice to help increase bandwidth is to add more fibers to the network; however, that is an expensive solution. Part of the research investment is focusing on switching the data between data centers and clients in the optical domain using circuit-switched networks [2].

By going back in time to the origin of optical switching network research, we would notice that the majority of research focuses on space switching. Switching in the space domain refers to the operation of switching the signal from one physical fiber to another. Meanwhile, time and wavelength domains have received a reasonable amount of attention. Switching in wavelength domain is a common practice, nowadays. However, switching Optical Time-Division Multiplexed (OTDM) signals in the optical domain has not been commercialized.

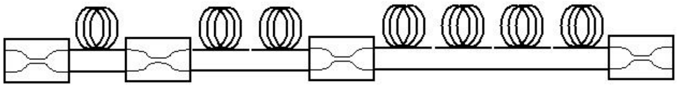


Figure 1. A delay element for 4 timeslots per frame

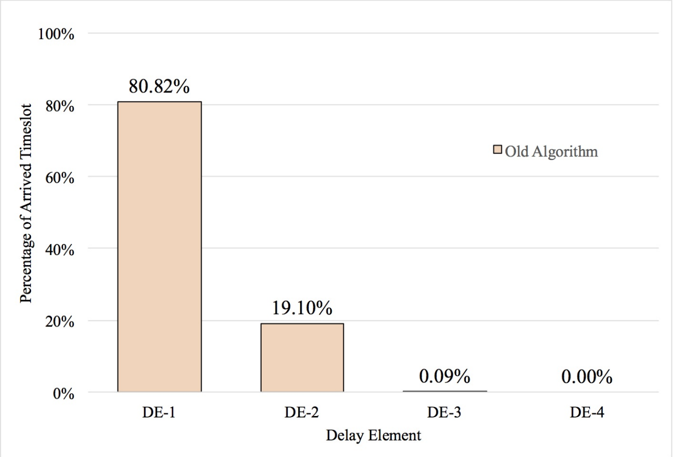


Figure 2. Results from the old control algorithm for 4 timeslots per frame [4]

In OTDM, channels are divided based on time. For the purpose of this paper, we assume that time channels (presented by timeslots) have a fixed duration of time. In addition, this paper presents a circuit switched network with call setup before starting data transmission. For better network utilization, an OTDM network does not require timeslot continuity, in which there has to be at least one timeslot vacant in every link to establish the connection.

If timeslot continuity is required, the exact timeslot index has to be vacant in every hop to establish the connection.If timeslot continuity is not required, a Photonic Timeslot Interchanger (PTSI) should be used at every node. Timeslot interchanging is the operation of switching timeslots with each other. Studies have proven that PTSIs improve system utilization and reduce network probability blocking [5][6].However, none of these studies have used the exact PTSI that was proposed by Thompson [7]. Building a PTSI consists of a number of Delay Elements (DE) connected together. The structure of a DE is presented in Figure 1.

The rest of this paper is organized as follows: Section II describe the motivation behind this work. The proposed control algorithm is described in Section III. The results are presented in Section IV. Finally, the conclusion and future works are presented in Sections V and VI, respectively.

## II. Motivation

The result from the first control algorithm [4] that controls such a PTSI shows that some of the DEs have been utilized less than 1% of the time, as shown in Figure 2.

Hence, we assume that if we make the algorithm more sophisticated, we might obtain better results and eliminate additional DEs. There are a number of benefits to reducing the number of DEs in the PTSI including:

- Reducing the number of hardware components in the fabric increases the overall availability [8].
- Adding more hardware to the fabric increases the cost, footprint and control algorithm complexity. A simple control algorithm reduces the fabric development and manufacturing time [8].
- As the light beam passes through optical components, it suffers from losses, mainly insertion loss. Hence, reducing the number of components that the signal passes through improves the signal strength.

## III. Proposed Model

Building a PTSI has been discussed in great detail in Thompson's initial study [7] and in Hasnawi and Thompson [4]. There has not been any change to the PTSI components. In this paper, our focus is to improve the control algorithm to eliminate as many additional DEs as possible. PTSI consists of DEs similar to the one in Figure 1. DEs are connected to form a binary tree on both sides, as in Figure 3. In this paper, we will generalize the switching components (splitters, switches, and combiners) using the term *Switching Module* (SWM). We assumed that the links between SWMs are perfect connectors with no loss or delay.

### A. Switching Assignment Matrix

A Switching Assignment (SWA) is defined as permuting input timeslots with output timeslots. Each input timeslot, $S_i^{in}$ , may be switched into *TS* possible output timeslots $S_j^{out}$. There are $TS$! possible SWAs. All possible SWAs form a *[TS!][TS]* matrix. Each row is indexed from *0* to *TS!-1* . Every SWA is presented in cyclic notation. For example: SWA (a)(bc) is read as Timeslot (a) at frame *F-1* is assigned to be switched to Timeslot (a) in the next frame, *F*, while Timeslot (b) and (c) at frame *F-1* are assigned to be switched to Timeslots (c) and (b) at frame *F*, respectively. The SWA index is assigned to the simulation during the initialization of each run. Since there are *M = 2* frames processed for every run, this study has two cases:

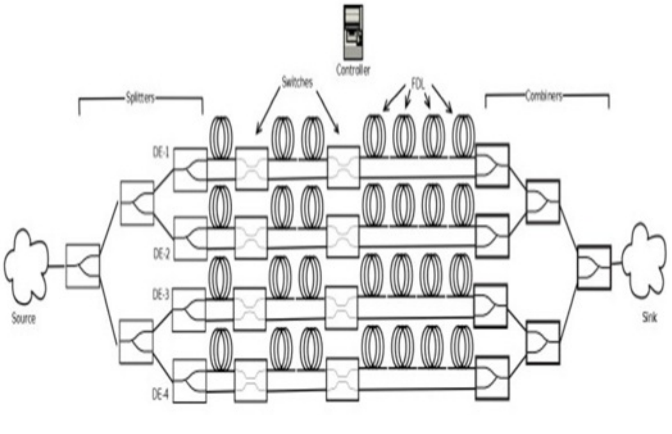


Figure 3. A complete PTSI for 4 timeslots per frame

- Case 1: Static Switching Assignment: both frames have the same switching assignment. Thus, there are (*TS!*) total number of possible permutations.
- Case 2: Dynamic Switching Assignments. Frames are independent from each other. Hence, Timeslots in the frame *F-1* should have independent SWA from *F*. In this case, there are *(TS!)M* total number of Switching Assignments.

In this study, we are going to apply every possible SWA for TS = 2, 4 and 8 for static cases. In addition, for the dynamic case, we are trying every SWA for TS = 2 and 4. We will omit TS = 8 for the dynamic switching assignment because the total number of runs required for every possible permutation would be greater than 1.6 billion.

TABLE I. Sample of the switching assignment for TS=4.

| SW Index | Input Timeslot | | | |
|---|---|---|---|---|
| | $S_0$ | $S_1$ | $S_2$ | $S_3$ |
| 0 | 0 | 1 | 2 | 3 |
| 1 | 0 | 1 | 3 | 2 |
| 2 | 0 | 2 | 1 | 3 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| 23 | 3 | 2 | 1 | 0 |

### B. Delay Matrix

Considering a synchronized system, every SWM must be pre-set during guard time, prior to the timeslot. Any misbehavior may affect the overall system and result in system failure. Some such scenarios include:

- If a timeslot arrives at SWM that is pre-set to an undesired switching state, the timeslot will exit at an undesired port.
- If two timeslots arrive at a switch simultaneously, each requiring a different switching state, at least one will be switched to an undesired output.
- If two timeslots arrive at a combiner simultaneously, at least one timeslot will be blocked.

Every SWM must be reserved (set to busy) and pre-set depending on the delay required for the timeslot. Computing the delay required to switch timeslots $S_i^{in}$ to $S_j^{out}$ is given by $D = TS + j - i$ ; where *j* is the timeslot output index and *i* is the timeslot input index. For example, assuming TS = 4 and SW index = 2 from Table I, we can extract that the cyclic notation for this assignment is ($S_0^{in}$)($S_1^{in}$ $S_2^{in}$)($S_3^{in}$). Using the delay equation, $S_0^{in}$ and $S_3^{in}$ each must be delayed by a duration equivalent to 4 timeslots, while $S_1^{in}$ must be delayed by 5 timeslots, and $S_2^{in}$ must be delayed by only 3. The required switching operation, as well as the required delay per timeslot, is illustrated in Figure 4. This first stage of the PTSI is the splitter stage, as in Figure 3, which forms a perfect binary tree with a total number of leaves $l = TS$ (equal to the number of DEs), and the depth of the tree given by $d = log_2 l + 1$. Thus, each timeslot passes through d splitters before it enters a DE. The combiner stage is identical to the splitter stage but on the other side of the fabric. Splitters do not perform interchange operations, thus, no delay at this stage, as seen in Table II.

However, switches and combiners do require delays before they change their status. Going back to the example in the

TABLE II. DELAY MATRIX FOR TS=4

| | Splitters Stage | | | Switches Stage | | Combiners Stage | | |
|---|---|---|---|---|---|---|---|---|
| | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| Delay 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Delay 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| Delay 2 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 |
| Delay 3 | 0 | 0 | 0 | 1 | 3 | 3 | 3 | 3 |
| Delay 4 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 |
| Delay 5 | 0 | 0 | 0 | 1 | 1 | 5 | 5 | 5 |
| Delay 6 | 0 | 0 | 0 | 0 | 2 | 6 | 6 | 6 |
| Delay 7 | 0 | 0 | 0 | 1 | 3 | 7 | 7 | 7 |

previous paragraph, switching $S_0^{in}$ to $S_0^{out}$ requires a delay with a duration of 4 timeslots. Thus, every splitter and switch stage in Figure 3 must change its stage during the guard time prior to the timeslot. The combiner stage can be set to busy and change its switching state during the guard time, prior to the timeslot transmission, until the timeslot arrives. However, doing so will prevent other timeslots from using these combiners from time t=0 to t=4. Therefore, to better utilize the fabric, the combiner stage will change its state after a delay of 4 timeslots.

The delay required for each SWM to change its state while efficiently utilizing the fabric is presented by a $[D_{max}][totalNumberOfSWMinThePath]$ matrix, as shown in Table II. $D_{max}$ is defined as the maximum delay required to interchange the first timeslot, $S_0^{in}$ , with the last timeslot, $S_{TS}^{in}$ , and is given by $D_{max} = 2TS - 1$.

The start holding time (*startHoldingTime*) is a parameter used in the simulation. It is a *timeStamp* at which the SWM should start holding its current state before releasing to the idle state and being set to free. Using Table II, $startHoldingTime = [currentSimulationTime] + [TS_{dur} * Delay]$. Once a timeslot exits the SWM, then SWM is set to free. For M=2 and TS=4, the initial module [7] utilizes each DE one fourth of the time.

### C. Select Path Algorithm

Select Path algorithm is what distinguishes this work from others. This algorithm is responsible for switching timeslots to the optimum DE in the PTSI. The first proposed PTSI model did not implement a path selection (or routing) algorithm.

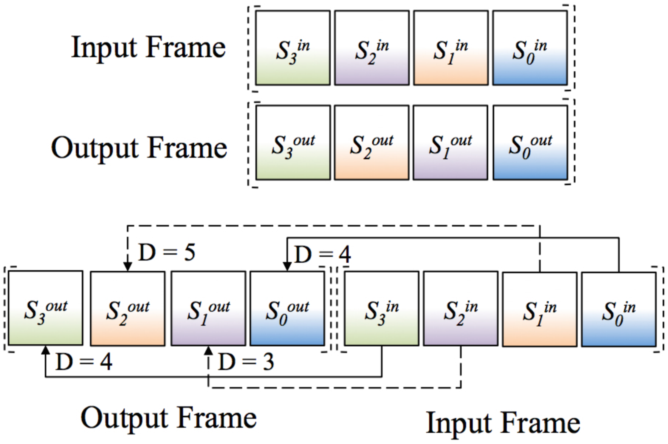
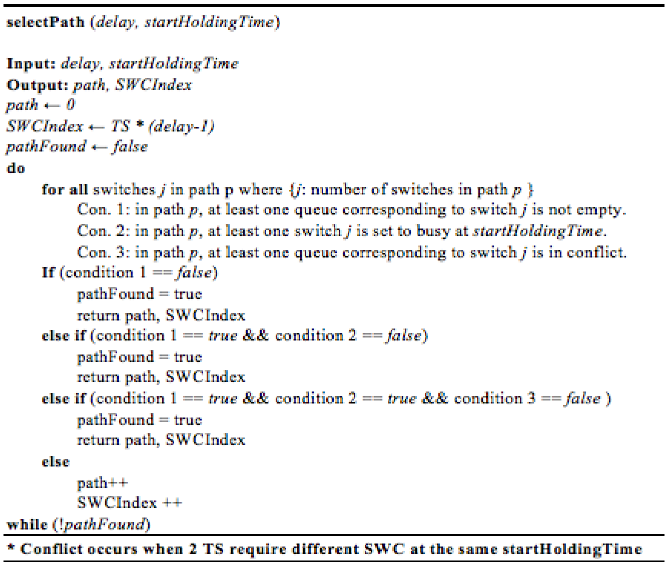


Figure 4. A graphical representation of PTSI operation for SWA index 2 with the required delay per timeslot.



```
selectPath (delay, startHoldingTime)

Input: delay, startHoldingTime
Output: path, SWCIndex
path ← 0
SWCIndex ← TS * (delay-1)
pathFound ← false
do
    for all switches j in path p where {j: number of switches in path p }
        Con. 1: in path p, at least one queue corresponding to switch j is not empty.
        Con. 2: in path p, at least one switch j is set to busy at startHoldingTime.
        Con. 3: in path p, at least one queue corresponding to switch j is in conflict.
    If (condition 1 == false)
        pathFound = true
        return path, SWCIndex
    else if (condition 1 == true && condition 2 == false)
        pathFound = true
        return path, SWCIndex
    else if (condition 1 == true && condition 2 == true && condition 3 == false )
        pathFound = true
        return path, SWCIndex
    else
        path++
        SWCIndex ++
while (!pathFound)
* Conflict occurs when 2 TS require different SWC at the same startHoldingTime
```

Figure 5. Select path algorithm

It assumed that blocking was avoided by switching $S_i^{in}$ to $DE_i$. Hence, every row should be utilized $M/(M * TS)$ of the time. This result led us to build a new path selection algorithm that would reduce the number of DEs in the PTSI, and increase the utilization to the maximum while maintaining zero probability of blocking.

The first select path algorithm was published by Hasnawi [4]. The algorithm did not allow 2 timeslots to travel through a switch simultaneously even if both required the same switching state.

The improved algorithm differs from the previously published algorithm by allowing 2 timeslots to pass through a switch simultaneously, if, and only if, both required the same switching state. The algorithm is presented in Figure 5.

### D. Switching Control

Switching Control (SWC) is a control signal that travels from the controller to the SWM. The purpose of this signal is to change the switching state of SWM. Switching control contains three fields:

- Switching State (Bool): changes the state of an SWM to *BAR* (0) or *CROSS* (1).
- Busy (Bool): if the SWM is reserved, then it is set to busy (1); otherwise, it is set to free (0).
- *startHoldingTime* (double): is the time at which the SWM should be switched to the new state and set to busy.
- *releaseTime* (double): is the time at which the SWM releases its current state and is set to free.

We assume that every SWM is a Lithium Niobate ($LiNbO_3$) directional coupler, which only requires a simple SWC, presented as a voltage. Hence, only the switching state is passed from the controller to SWM and the rest of the fields are used inside the controller.

Once a path is selected, the controller calls another algorithm to reserve the path for the incoming timeslot. The algorithm is presented in Figure 6.

### E. Insert Switching Control

Every SWM has a SWC queue at the controller. Switching Controls are inserted in order, based on the $startHoldingTime$. If two or more SWC for a given SWM have the same switching state at $startHoldingTime$, then only one SWC will be stored in the queue and the rest will be discarded.

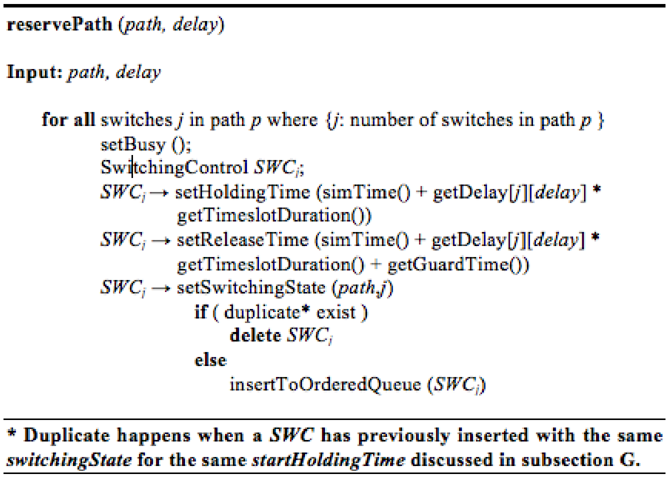
### F. Send Switching Control

At every guard time prior to any timeslot, each head of the queue is examined. If the current simulation time, $simTime$ equals $startHoldingTime$, then the SWC is popped and sent to its corresponding SWM; otherwise, the queue will be ignored and examined at the next guard time.

### G. Simulation Tool and Assumption

We used an Omnet++ [9] simulation tool to build every component in our PTSI. For each simulation run, we assumed the following:

- There were M=2 frames per run.
- Each frame had TS = 2, 4 or 8 timeslots.
- The size of each timeslot was fixed and equal to $10^6$ bits.
- The data rate equaled $R = 10^9$ bps
- A Timeslot Duration ($TS_{dur}$) equals the timeslot size divided by the data rate.
- There was a guard time between consecutive timeslots. This guard time equaled the SWM switching speed.
- Timeslots SWAs' follow the permutation table, discussed in subsection A.
- There is frame integrity, in which timeslots are switched within the boundary of the frame.
- Each run starts with an empty fabric; then, timeslots are generated at the deterministic rate of $TS_{dur}$.
- Every channel is a point-to-point connection.
- Each simulation run is independent. For each run, we verified that each timeslot generated by the source was received at the destination.



```
reservePath (path, delay)

Input: path, delay

    for all switches j in path p where {j: number of switches in path p }
            setBusy ();
            SwitchingControl $SWC_j$;
            $SWC_j$ → setHoldingTime (simTime() + getDelay[j][delay] *
                      getTimeslotDuration())
            $SWC_j$ → setReleaseTime (simTime() + getDelay[j][delay] *
                      getTimeslotDuration() + getGuardTime())
            $SWC_j$ → setSwitchingState (path,j)
                   if ( duplicate* exist )
                        delete $SWC_j$
                   else
                        insertToOrderedQueue ($SWC_j$)
```

* Duplicate happens when a *SWC* has previously inserted with the same *switchingState* for the same *startHoldingTime* discussed in subsection G.

Figure 6. Reserve path algorithm

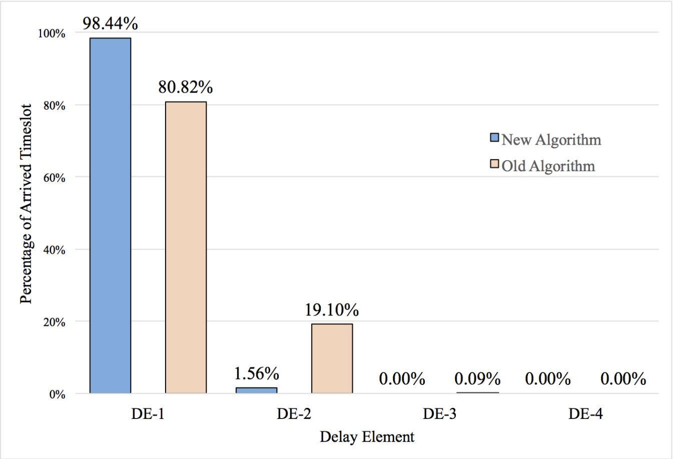


Figure 7. Comparing the old and new Select Path algorithm for TS=4. Each bar represents the utilization of the DE

## IV. Results

The new algorithm provides a significant improvement. We were able to further reduce the number of DEs for every case. The results show that for 2, 4, and 8, the number of DEs required to provide non-blocking PTSI is 1, 2, and 2, respectively. Note that, in this study, we only simulated the static case for TS=8 and we were able to interchange every timeslot with only two DEs.

The number of DEs required to interchange TS = 4 timeslots per frame using the improved algorithm is illustrated in Figure 7. Similarly, the number of DEs required to interchange TS = 8 timeslots per frame for the static case using the improved algorithm is illustrated in Figure 8.

From a software perspective, reducing the size of a PTSI results in easier and faster control. Searching for a path from three available paths is faster than searching from eight.

Lastly, eliminating half the DEs from a PTSI for TS = 4, not only reduced the footprint size, but also improved the received signal power at the destination node. Each timeslot passed through 6 SWMs instead of 8, and passed through 9 timeslots instead of 11 for TS = 8. The average insertion loss for SWM was between 4 and 5dB for wavelength =1550 [10][11][12]. Hence, reducing the PTSI size improved the received power signal by an average of $10dB$ for TS = 4 and TS = 8.

## V. Conclusion

In this work, we were able to further reduce the number of DEs in a PTSI. The algorithm improvement focused on additional reductions on the number of DEs while maintaining the same code complexity.The impact of this additional improvement is a scalable system. The number of channels (presented in timeslots) does not increase proportionally with the size of the PTSI. We were able to reduce an additional 25% of the number of DEs for TS = 4, compared with the previous work [4].In addition, for TS = 8, we reduced the number of DEs by an additional 12.5%, compared with the previous work. Both works show that the dynamic case requires one additional DE than the static case to provide non-blocking interchanging. If this statement is generally true, then for TS = 8, the

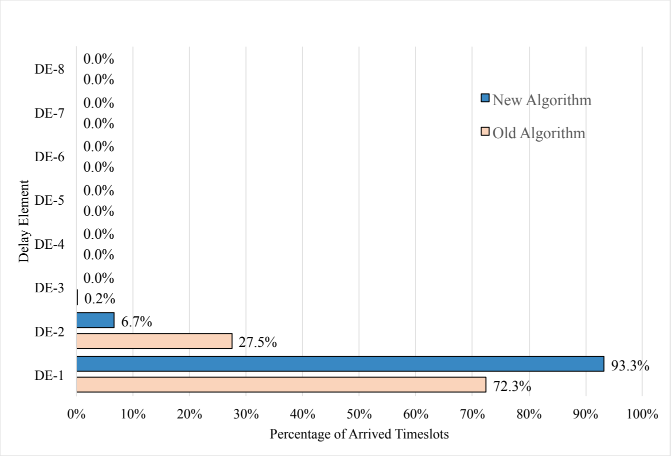


Figure 8. Comparing the old and new Select Path algorithm for TS=8 (static case). Each bar represents the utilization of the DE

required number of DEs to provide non-blocking switching is 3. These six scenarios (TS = 2, 4, and 8 for both static and dynamic cases), result in reducing the number of DEs required for non-blocking timeslot interchanging to $log_2(TS)$; however, it needs to be proven for $TS = 2^N$. The improvement on the number of DEs from the initial work [7], on the first (old) control algorithm [4], to the second (improved) control algorithm is presented in Figure 9. Additional benefits from this improvement include reducing the footprint, circuit temperature, power loss and monetary cost.

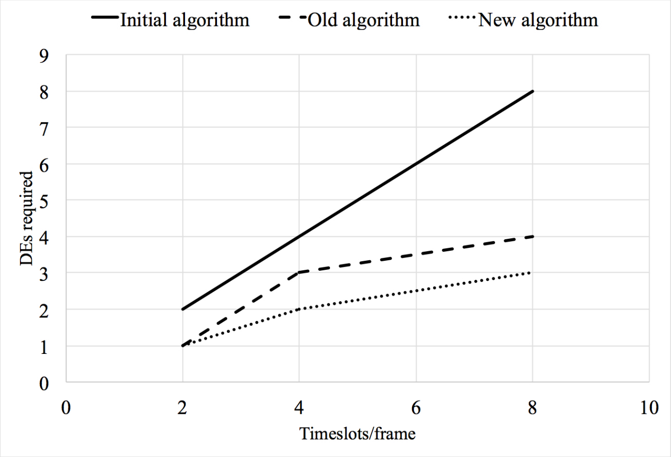


Figure 9. PTSI improvement over time based on the number of DEs

## VI. Future Work

This simulation starts with an empty network and then the source starts generating timeslots until the network reaches 100% utilization. The work will be tested on a testbed network, such as NSFNet under different loads. In addition, this work could extend to larger number of timeslots per frame such as 64, 128, . $2^n$.

## Acknowledgment

The authors would like to thank Taibah University for supporting and partially sponsoring this work. We would like to thank Rakan and Anh for help in debugging and improving the code.

## References

[1] C. Develder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester, "Optical networks for grid and cloud computing applications," Proceedings of the IEEE, vol. 100, no. 5, 2012, pp. 1149–1167.

[2] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," Communications Surveys & Tutorials, IEEE, vol. 14, no. 4, 2012, pp. 1021–1036.

[3] C. V. N. Index, "The zettabyte era–trends and analysis," Cisco white paper, 2013.

[4] L. E. Hasnawi and R. A. Thompson, "Photonic timeslot interchangers with a reduced number of feed-forward fiber delay lines," Procedia Computer Science, vol. 34, 2014, pp. 47–54.

[5] J. Yates, J. Lacey, and D. Everitt, "Blocking in multiwavelength tdm networks," Telecommunication Systems, vol. 12, no. 1, 1999, pp. 1–19.

[6] Y. C. Huei, P. H. Keng, and N. Krivulin, "Average network blocking probabilities for tdm wdm optical networks with otsis and without wc," in Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS'07. 15th International Symposium on. IEEE, 2007, pp. 424–431.

[7] R. A. Thompson, "Optimizing photonic variable-integer-delay circuits," in Photonic Switching. Springer, 1988, pp. 158–166.

[8] M. Zdeblick, "Design variables prevent a single industry standard," Laser Focus World, vol. 37, no. 3, 2001.

[9] "Omnet++ discrete event simulator," http://www.omnetpp.org/, accessed: 2015-03-01.

[10] X. Ma and G.-S. Kuo, "Optical switching technology comparison: optical mems vs. other technologies," Communications Magazine, IEEE, vol. 41, no. 11, 2003, pp. S16–S23.

[11] "Coralign low loss moving fiber optical switches," http://luminos.com/products/switches/downloads/switch_datasheet_detail.pdf, accessed: 2015-03-01.

[12] "Jdsu 2x2 interferometric switch," http://www.jdsu.com/en-us/Optical-Communications/Products/a-z-product-list/Pages/switch-2x2-interferometric-lithium-niobate.aspx#.VP0albPF840, accessed: 2015-03-01.

# The Uncompress Application on Distributed Communications Systems

Sergio De Agostino
Computer Science Department
Sapienza University
Rome, Italy
Email: deagostino@di.uniroma1.it

***Abstract*—The most popular compressors are based on Lempel-Ziv coding methods. Zip compressors and Unzip decompressors apply the sliding window method, while other applications as Compress and Uncompress under Unix and Linux platforms use the so-called LZW compressor and decompressor. LZW compression is less effective but faster than the zipping applications. We face the problem of how to implement Lempel-Ziv data compression on today's large scale distributed communications systems. Zipping and unzipping files is parallelizable in theory. However, the number of global computation steps is not bounded by a constant and a local computation approach is more advantageous on a distributed system. Such approach might cause a lack of robustness when scalability properties are required. Differently from the Zip compressors, the LZW encoder/decoder presents an asymmetry with respect to global parallel computation since the encoder is not parallelizable while the decoder has a very efficient parallelization. We show that, in practice, the number of iterations of the LZW parallel decoder (Uncompress) is much less than ten units. Since scalability and robustness are generally guaranteed if bounding the number of global computation steps is possible, LZW is more attractive than Zip for distributed communications systems in those cases (which are the most common in practice) where compression is performed only once or very rarely, while the frequent reading of raw data needs fast decompression.**

***Keywords-distributed application; communication; scalability; robustness; data compression***

## I. INTRODUCTION

The most popular compressors are based on Lempel-Ziv coding methods (LZ compression). Zip compressors and Unzip decompressors apply the sliding window method, while other applications as Compress and Uncompress under Unix and Linux platforms use the so-called LZW compressor and decompressor. LZW compression is less effective but faster than the zipping applications. LZ compression [1][2] is based on string factorization. Two different factorization processes exist with no memory constraints. With the first one (LZ1) [1], each factor is independent from the others since it extends by one character the longest match with a substring to its left in the input string. With the second one (LZ2) [2], each factor is instead the extension by one character of the longest match with one of the previous factors. This computational difference implies that while LZ1 compression has theoretical parallel algorithms [3]-[6], LZ2 compression is hard to parallelize [7]. This difference is mantained when bounded memory versions of LZ compression are considered [5][6][8]. On the other hand, parallel decompression is possible for both approaches [4][9]. The Zip compressors implement a bounded memory version (sliding window) of LZ1, while LZW compression is a bounded memory variant of LZ2.

We face the problem of how to implement LZ data compression on today's large scale distributed communications systems. The computing techniques involved in the design of parallel and distributed algorithms strictly relate to the computational model on which the distributed communications system is based. The efficiency of a technique designed for a specific model can consistently deteriorates when applied to a different system. This is particularly evident when a technique designed for a shared memory parallel random access machine (PRAM) is implemented on a distributed memory system. Indeed, when the system is scaled up the communication cost is a bottleneck to linear speed-up. So, we need to limit the interprocessor communication either by involving more local computation or by bounding the number of global computation steps in order to obtain a practical algorithm. Local computation might cause a lack of robustness when scalability properties are required. On the other hand, scalability and robustness are generally guaranteed if bounding the number of global computation steps is possible for a specific problem.

As mentioned above, zipping and unzipping files are parallelizable in theory. However, the number of global computation steps is not bounded by a constant and the local computation approach is more advantageous on a distributed memory system. A distributed algorithm, approximating in practice the compression effectiveness of the Zip application, has been realized in [10] on an array of processor with no interprocessor communication. An approach using a tree architecture slightly improves compression effectiveness [11]. Yet, distributed algorithms approximating in practice the compression effectiveness of the Compress application have been realized in [12] with very low communication cost. However, scalability and robustness for each of these LZ compression distrbuted algorithms are guaranteed only on very large size files.

In this paper, we evidentiate that, differently from the Zip compressors, the LZW encoder/decoder presents an asymmetry with respect to global parallel computation since the encoder is not parallelizable while the decoder has a very efficient parallelization. We show that, in practice, the number of iterations of the LZW parallel decoder is much less than ten units. This makes LZW more attractive than Zip for distributed communications systems in those cases (which are the most common in practice) where compression is performed very rarely while the frequent reading of raw data needs fast decompression.

In Section II, we describe the LZ data compression techniques and their bounded memory versions. In Section III, we discuss the parallel complexity of LZ data compression and decompression. Section IV shows how the implementation of the parallel LZW decompressor (Uncompress) is suitable for distributed communications systems. Conclusions and future work are given in Section V.

## II. LZ Data Compression

LZ data compression is a dictionary-based technique. Indeed, the factors of the string are substituted by *pointers* to copies stored in a dictionary which are called *targets*. LZ1 (LZ2) compression is also called the sliding (dynamic) dictionary method.

### *A. LZ1 Compression*

Given an alphabet $A$ and a string $S$ in $A^*$, the LZ1 factorization of $S$ is $S = f_1 f_2 \cdots f_i \cdots f_k$, where $f_i$ is the shortest substring which does not occur previously in the prefix $f_1 f_2 \cdots f_i$ for $1 \leq i \leq k$. With such factorization, the encoding of each factor leaves one character uncompressed. To avoid this, a different factorization was introduced (LZSS factorization) where $f_i$ is the longest match with a substring occurring in the prefix $f_1 f_2 \cdots f_i$ if $f_i \neq \lambda$, otherwise $f_i$ is the alphabet character next to $f_1 f_2 \cdots f_{i-1}$ [13]. $f_i$ is encoded by the pointer $q_i = (d_i, \ell_i)$, where $d_i$ is the displacement back to the copy of the factor and $\ell_i$ is the length of the factor (LZSS compression). If $d_i = 0$, $l_i$ is the alphabet character. In other words, the dictionary is defined by a window sliding its right end over the input string, that is, it comprises all the substrings of the prefix read so far in the computation. It follows that the dictionary is both *prefix* and *suffix*, since all the prefixes and suffixes of a dictionary element are dictionary elements.

### *B. LZ2 Compression*

The LZ2 factorization of a string $S$ is $S = f_1 f_2 \cdots f_i \cdots f_k$, where $f_i$ is the shortest substring which is different from all the previous factors. As for LZ1, the encoding of each factor leaves one character uncompressed. To avoid this, a different factorization was introduced (LZW factorization) where each factor $f_i$ is the longest match with the concatenation of a previous factor and the next character [14]. $f_i$ is encoded by a pointer $q_i$ to such concatenation (unbounded LZW compression). Differently from LZ1 and LZSS, the dictionary is only prefix.

### *C. Bounded Size Dictionary Compression*

The factorization processes described in the previous subsections are such that the number of different factors (that is, the dictionary size) grows with the string length. In practical implementations instead the dictionary size is bounded by a constant and a fixed length code is used for the pointers. With sliding window compression, this can be simply obtained by using a fixed length window (therefore, the left end of the window slides as well) and by bounding the match length. Simple real time implementations are realized by means of hashing techniques providing a specific position in the window where a good apprixmation of the longest match is found on realistic data. The window length is usually several kilobytes. The compression tools of the Zip family, as the Unix command gzip for example, use a window size of at least 32 Kb.

With LZW compression the dictionary elements are removed by using a deletion heuristic [15]. Let $d + \alpha$ be the cardinality of the fixed size dictionary where $\alpha$ is the cardinality of the alphabet. With the FREEZE deletion heuristic, there is a first phase of the factorization process where the dictionary is filled up and “frozen”. Afterwards, the factorization continues in a “static” way using the factors of the frozen dictionary. In other words, the LZW factorization of a string $S$ using the FREEZE deletion heuristic is $S = f_1 f_2 \cdots f_i \cdots f_k$, where $f_i$ is either the longest match with the concatenation of a previous factor $f_j$, with $j \leq d$, and the next character or the current alphabet character if there is no match. The shortcoming of the FREEZE heuristic is that, after processing the string for a while, the dictionary often becomes obsolete. A more sophisticated deletion heuristic is RESTART, which monitors the compression ratio achieved on the portion of the input string read so far and, when it starts deteriorating, restarts the factorization process (*standard LZW encoding*). Let $f_1 f_2 \cdots f_j \cdots f_i \cdots f_k$ be such a factorization with $j$ the highest index less than $i$ where the restart operation happens. Then, $f_j$ is an alphabet character and $f_i$ is the longest match with the concatenation of a previous factor $f_h$, with $h \geq j$, and the next character (or the current alphabet character if there is no match since the restart operation removes all the elements from the dictionary but the alphabet characters). This heuristic is the one used by standard applications (as Compress under Unix and Linux platforms) since it has a good compression effectiveness and it is easy to implement. Usually, the dictionary performs well in a static way on a block long enough to learn another dictionary of the same size. This is what is done by the SWAP heuristic. When the other dictionary is filled, they swap their roles on the successive block.

The best deletion heuristic is the least recently used (LRU) strategy. The LRU deletion heuristic removes elements from the dictionay in a “continuous” way by deleting at each step of the factorization the least recently used factor, which is not a proper prefix of another one. In [8], a relaxed version (RLRU$p$) was introduced. RLRU$p$ partitions the dictionary in $p$ equivalence classes, so that all the elements in each class are considered to have the same "age" for the LRU strategy. RLRU$p$ turns out to be as good as LRU even when $p$ is equal to 2 [16]. Since RLRU2 removes an arbitrary element from the equivalence class with the "older" elements, the two classes can be implemented with a couple of stacks, which makes RLRU2 slightly easier to implement than LRU in addition to be more space efficient. SWAP is the best heuristic among the "discrete" ones. Each of the bounded versions of LZW compression, described in this subsection, can be implemented in real time by storing the dictionary in a trie data structure.

## III. Parallel Complexity

The model of computation we consider in this section is the CREW (cuncurrent read, exclusive write) PRAM, that is, a parallel machine where processors access a shared memory without writing conflicts. As mentioned in the introduction, speed is more relevant with decompression than with compression since in most cases compression is performed very rarely while the frequent reading of raw data needs a fast decoder. Therefore, we briefly discuss the parallel complexity issues concerning compression and, then, describe the parallel decoders for sliding window compression and LZW compression.

### *A. Parallel Complexity of LZ Compression*

LZSS (or LZ1) compression (that is, the zipping application) can be efficiently parallelized from a theoretical point of view [3]-[6]. On the other hand, LZW (or LZ2) compression is P-complete [7] and, therefore, hard to parallelize. Decompression, instead, is parallelizable for both methods [4][9]. As far as bounded size dictionary compression is concerned, the "parallel computation thesis" claims that sequential work space and parallel running time have the same order of magnitude, giving theoretical underpinning to the realization of parallel algorithms for LZW compression using a deletion heuristic. However, the thesis concerns unbounded parallelism and a practical requirement for the design of a parallel algorithm is a limited number of processors. A stronger statement is that sequential logarithmic work space corresponds to parallel logarithmic running time with a polynomial number of processors. Therefore, a fixed size dictionary implies a parallel algorithm for LZW compression satisfying these constraints. Realistically, the satisfaction of these requirements is a necessary but not a sufficient condition for a practical parallel algorithm since the number of processors should be linear, which does not seem possible for the RESTART deletion heuristic (that is, Compress), while the SWAP heuristic does not seem to have a parallel decoder [8]. Moreover, the SC$^k$-hardness of LZ2 compression using the LRU deletion heuristic and a dictionary of polylogarithmic size shows that it is unlikely to have a parallel complexity involving reasonable multiplicative constants [8]. In [8], the RLRU$p$ relaxed version was introduced in order to obtain the first (and only so far) natural SC$^k$-complete problem by partitioning the dictionary in $p$ equivalence classes and by considering all the elements in each class to have the same "age" for the LRU strategy. As mentioned in the previous section, RLRU2 turns out to be as good as LRU and it is slightly easier to implement in addition to be more space efficient. In conclusion, the only practical LZW compression algorithm for a shared memory parallel system is the one using the FREEZE deletion heuristic [6].

### *B. Parallel Unzip*

The design of a parallel decoder for sliding window compression (that is, the unzipping application) is based on a reduction to the problem of finding the trees of a forest in O$(k)$ time with O$(n/k)$ processors on a CREW PRAM, if $k$ is $\Omega(\log n)$ and $n$ is the number of nodes. Given the sequence of pointers $q_i = (d_i, \ell_i)$, for $1 \leq i \leq m$, produced by the application zipping an input file, let $s_1, ..., s_m$ be the partial sums of $l_1, ..., l_m$. Then, the target of $q_i$ encodes the substring over the positions $s_{i-1} + 1 \cdots s_i$ of the output string. Link the positions $s_{i-1} + 1 \cdots s_i$ to the positions $s_{i-1} + 1 - d_i \cdots s_{i-1} + 1 - d_i + l_i - 1$, respectively. If $d_i = 0$, the target of $q_i$ is an alphabet character and the corresponding position in the output string is not linked to anything. Therefore, we obtain a forest where all the nodes in a tree correspond to positions of the decoded string where the character is represented by the root. The reduction from the decoding problem to the problem of finding the trees in a forest can be computed in O$(k)$ time with O$(n/k)$ processors where $n$ is the length of the output string, because this is the complexity of computing the partial sums since $m \leq n$. Afterwards, O$(n/k)$ processors store the parent pointers in an array of size $n$ for blocks of $k$ positions and apply the pointer jumping technique to find the trees.

### *C. Parallel Uncompress*

We already pointed out that the decoding problem is interesting independently from the computational efficiency of the encoder. This is particularly evident in the case of compressed files stored in a ROM since only the computational efficiency of decompression is relevant. With the RESTART deletion heuristic, a special mark occurs in the sequence of pointers each time the dictionary is cleared out so that the decoder does not have to monitor the compression ratio (Uncompress application). The positions of the special mark can be detected by parallel prefix.

Let $Q_1 \cdots Q_N$ be the standard LZW encoding of a string $S$, drawn over an alphabet $A$ of cardinality $\alpha$, with $Q_h$

sequence of pointers between two consecutive restart operations, for $1 \leq h \leq N$. Let $D$ be the bounded size dictionary employed by the compression algorithm, with $d = |D|$. Each $Q_h$ can be decoded independently. Let $q_1...q_m$ be the sequence of pointers $Q_h$ encoding the substring $S'$ of $S$. The decoding of $Q_h$ can be parallelized on a CREW PRAM in O($(log(L))$ time with O($|S'|$) processors, where $L$ is the maximum length of a pointer target. $d$ is a theoretical upper bound to $L$, that is tight for unary strings. The algorithm works with an initially null $m$ x $d$ matrix $M$ and applies to $M$ the procedure of Figure 1 [9].

```
$k = 1$;
in parallel for $1 \leq i \leq m$ do begin
    $M[1, i] := q_i$;
    $last[i] := 1$;
    $value[i] := M[1, i] - d$;
    while $value(i) > 0$ do begin
        in parallel for $1 \leq j \leq k$ do
            if $j \leq last(value(i))$ then $M[last(i) + j, i] := M[j, value(i)]$;
        $last(i) := last(i) + last(value(i))$;
        $value(i) := value(value(i))$;
        $k = 2k$;
    end;
end
```

Figure 1. The CREW PRAM procedure.

At each step, $last[i]$ is the last nonnull component on the $i^{th}$ column considered. The nonnull components of the $value(i)^{th}$ column are copied on the $i^{th}$ column in the positions after $last[i]$. Note that $value(i)$ is strictly less than $i$. Then, $value(i)$ is updated by setting $value(i) := value(value(i))$. The iteration stops on the column $i$ when $value(i)$ is less or equal to zero. This procedure takes O($|S'|$) processors and O($log(L)$) time on a CREW PRAM, since the number of nonnull componenents on a column doubles at each step. The target of the pointer $q_i$ is the concatenation of the target of the pointer in position $q_i - \alpha$ with the first character of the target of the pointer in position $q_i - \alpha + 1$, since the dictionary contains initially the alphabet characters. At the end of the procedure, $M[last[i], i]$ is the pointer representing the first character of the target of $q_i$ and $last[i]$ is the target length. Then, we conclude that $M[last[M[j, i] - d + 1], M[j, i] - d + 1]$, for $1 \leq j \leq last[i] - 1$, is the pointer representing the $(last[i] - j + 1)^{th}$ character of the target of $q_i$. That is, we have to look at the pointer values written on column $i$ and consider the last nonnull components of the columns in the positions given by such values decreased by $\alpha - 1$. Such components must be concatenated according to the bottom-up order of the respective values on column $i$. By mapping each component into the correspondent alphabet character, we obtain the suffix following the first character of the target of $q_i$ and the pointers are, therefore, decoded.

## IV. Distributed Communications Systems and The Uncompress Application

Shared memory machines, as the PRAM model, are ideal systems for distributed communications. Realistically, such systems are feasible with the current technology only when the scale is very limited (ten units is the order of magnitude). The scalability requirement implies that the memory of the system is distributed. However, the PRAM model might be useful for a first approach to the design of an algorithm for distributed communications systems. Before discussing such approach, we consider distributed memory systems with no or very low interprocessor communication cost during the computational phase and, then, we discuss the requirements that a model of computation must have in order to yield a practical algorithm for distributed communications systems. Finally, we discuss the implementation of Uncompress and compare it with Unzip.

### A. Star and Extended Star Networks

Distributed memory systems have two types of complexity, the interprocessor communication and the input-output mechanism. While the input/output issue is inherent to any parallel algorithm and has standard solutions, the communication cost of the computational phase after the distribution of the data among the processors and before the output of the final result is obviously algorithm-dependent. So, as mentioned in the introduction, we need to limit the interprocessor communication either by involving more local computation or by bounding the number of global computation steps in order to design a practical algorithm. If we consider the local computation approach. the simplest model is a simple array of processors with no interconnections and, therefore, no communication cost. Such array of processors could be a set of neighbors linked directly to a central node (from which they receive blocks of the input) to form a so called *star* network (a rooted tree of hn height 1). In an *extended* star, each node adjacent to the central one has a set of leaf neighbors (a rooted tree of height 2). Such extension is useful in practice to scale up the system.

For every integer $k$ greater than 1, we can apply in parallel sliding window compression to blocks of length $kw$ with O($n/kw$) processors connected to a central node of a star network, where $n$ and $w$ are the lengths of the input string and the window respectively [10]. If the order of magnitude of the block length is greater than the one of the window length, the compression effectiveness of the distributed implementation is about the same as the sequential one on realistic data. Since the compression tools of the Zip family use a window size of at least 32 Kb, the block length should be about 300 Kb and the file size should be about one third of the number of processors in megabytes. Therefore, the application is suitable only for a small scale system unless the file size is very large.

As far as LZW compression is concerned, if we use a RESTART deletion heuristic clearing out the dictionary every $\ell$ characters of the input string we can trivially parallelize the factorization process with an O$(\ell)$ time, O$(n/\ell)$ processors distributed algorithm. In order to speed up the static phase after the dictionary is filled up, an implementation is provided on an extended star topology in [12]. The Compress application employs a dictionary of size $2^{16}$ and works with the RESTART deletion heuristic (also, called LZC compression [17]). The block length needed to fill up a dictionary of this size is approximately 300 Kb and the scalability and robustness issues with Compress are the same as with Zip.

### B. A Model of Computation

Distributed communications systems allow global computation and bounding the number of computational steps is a requirement for the design of a practical algorithm. In [18], necessary requirements for a practical model of distributed computation were proposed, by considering the MapReduce programming paradigm which is the most in fashion for the design of an application for distributed communications systems. The following complexity requirements are stated as necessary for a practical interest:

- the number of global computation steps is polylogarithmic in the input size $n$;
- the number of processors and the amount of memory for each processor are sublinear;
- at each step, each processor takes polynomial time.

In [18], it is also shown that a $t(n)$ time CREW PRAM algorithm using subquadratic work space and a subquadratic number of processors has an implementation satisfying the above requirements if $t(n)$ is polylogarithmic. Indeed, the number of global computation steps of the implementation is O$(t(n))$ while the subquadratic work space is partitioned among a sublinear number of processors taking polynomial computational time. Such requirements are necessary but not sufficient to guarantee a speed-up of the computation. Obviously, the total running time cannot be higher than the sequential one and this is trivially implicit in what is stated in [18]. The non-trivial bottleneck is the communication cost of the computational phase. This needs to be checked experimentally since the number of global computation steps can be polylogarithmic in the input size. The only way to guarantee with absolute robustness a speed-up with the increasing of the number of nodes is to design distributed algorithms implementable with no interprocessor communication. Moreover, if we want the speed-up to be linear then the total running time of a processor must be O$(t(n)/n^{1-\epsilon})$, where $t(n)$ is the sequential time and $n^{1-\epsilon}$ is the number of processors. These stronger requirements are satisfied by the distributed implementation of Zip and Unzip presented in the first subsection. The LZW encoder/decoder implemented on the extended star network has a low communication cost, which is still affordable. Based on a worst case analysis, a more robust approach with no interprocessor communication is presented in [19] for compressing very large size files, which uses the RLRU2 deletion heuristic.

Generally speaking, an application on distributed communications systems has a practical interest if the number of global computation steps is about ten units or less. This is obtained from the simulation of the CREW PRAM implementation of the Uncompress application, together with the other requirements mentioned above.

### C. Uncompress versus Unzip with Pointer Jumping

Computing the trees of a forest for Unzip and, implicitly, computing for each tree of a forest the paths from the nodes to the root for Uncompress are the problems we faced with parallel decompression. If we use a parent array as data structure to represent a forest, these problems can easily be solved on a CREW PRAM by running in parallel the pointer jumping operation *parent*$[i]$ = *parent*$[$*parent*$[i]]$. The procedure takes a number of processors linear in the number of nodes and a time logarithmic in the maximum height of a tree. However, while with the unzipping application the maximum height of a tree can reach the order of magnitude of the string length, Uncompress deals with very shallow trees. This means that we can parallelize the LZW decoder with many fewer iterations. Moreover, the number of children for each node is very limited in practice and cuncurrent reading can be easily managed by standard bdroadcasting techniques on today's available clusters.

Now, we are ready to discuss the complexity issues with respect to distributed communications systems of the parallelization of the Uncompress application presented in [9]. Standard LZW encoding applies the ”restart” operation to a dictionary of size $2^{16}$ in the Unix and Linux Compress applications, as pointed out in the first subsection, and similar applications have been realized with Stuffit on Windows and Dos platforms. As previously mentioned, the theoretical upper bound to the factor length is the dictionary size, which is tight in the unary string case. However, on realistic data we can assume that the maximum factor length $L$ is such that $10 < L < 20$. The motivation for this assumption is that, in practice, the maximum length of a factor is much smaller than the dictionary size. For example, when compressing english text with sixteen bits pointers, the average match length will only be about five units (for empirical results, see [15]). In some exceptional cases, the maximum factor length will reach one hundred units, that is, the number of iterations (global computation steps) will be equal to seven units. If $Q^1 \cdots Q^N$ is the encoding of the input string $S$, with $Q^h = q^h_1 ... q^h_{m^h}$ sequence of pointers between two consecutive restart operations for $1 \leq h \leq N$, let $H$ =max$\{m^h : 1 \leq h \leq N\}$. It is easy to implement the parallel procedure on a cluster of $H$ processors, where the input is a set $\cup_{h=1}^N V^h$ and each element in $V^h$ is a pointer $q^h_i$, for $1 \leq i \leq m^h$ and $1 \leq h \leq N$. This set is distributed among the $H$ processors so that the $i$-th processor receives pointer $q^h_i$ (if it exists), for $1 \leq h \leq N$.

The sub-linearity requirements stated in [18] are satisfied, since $N$ and $m^h$, for $1 \leq h \leq N$, are generally sub-linear in practice. Moreover, the running time multiplied by the number of processors is O$(T)$, with T sequential time, since the number of global computation steps is about ten units (optimality requirement). This makes the implementation of practical interest.

## V. Conclusion

In this paper, we presented an implementation of the Uncompress application on distributed communications systems. Although the Compress application is not parallelizable in practice, those characteristics implying hardness results with respect to the parallel complexity of the encoder are the same providing asymmetrically a practical parallel decoder. Since, in most cases, compression is performed only once or very rarely, while the frequent reading of raw data needs fast decompression, Compress can be considered attractive in a parallel fashion. Indeed, Unzip is much less practical to parallelize.

Parallel and distributed algorithms for LZ data compression and decompression is a field that has developed in the last twenty years from a theoretical approach concerning parallel time complexity with no memory constraints to the practical goal of designing distributed algorithms with bounded memory and low communication cost. However, there is a lack of robustness of the practical compression distributed algorithms when the system is scaled up and the order of magnitude of the file size is smaller than one gigabyte. As long as the communication cost is a relevent bottleneck of current technology, considering Compress and focusing only on speeding up Uncompress has good motivations since there is still a need for compression for purposes both of storage and transmission when the file size is a hundred megabytes or less and a novel application speeding up decoding requires robustness in such cases. As future work, we would like to implement Uncompress on today's large scale commodity clusters.

## References

[1] A. Lempel and J. Ziv, *A Universal Algorithm for Sequential Data Compression,* IEEE Transactions on Information Theory, vol. 23, 1977, pp. 337-343.

[2] J. Ziv and A. Lempel, *Compression of Individual Sequences via Variable-Rate Coding,* IEEE Transactions on Information Theory, vol. 24, 1978, pp. 530-536.

[3] M. Crochemore and W. Rytter, *Efficient Parallel Algorithms to Test Square-freeness and Factorize Strings,* Information Processing Letters, vol. 38, 1991, pp. 57-60.

[4] M. Farach and S. Muthikrishnan, *Optimal Parallel Dictionary Matching and Compression,* Proceedings SPAA, 1995, pp. 244-253.

[5] S. De Agostino, *Parallelism and Dictionary-Based Data Compression,* Information Sciences, vol. 135, 2001, pp. 43-56.

[6] S. De Agostino, *Lempel-Ziv Data Compression on Parallel and Distributed Systems,* Algorithms, vol. 4, 2011, pp. 183-199.

[7] S. De Agostino, *P-complete Problems in Data Compression,* Theoretical Computer Science, vol. 127, 1994, pp. 181-186.

[8] S. De Agostino and R. Silvestri, *Bounded Size Dictionary Compression:* SC$^k$*-Completeness and* NC *Algorithms,* Information and Computation, vol. 180, 2003, pp. 101-112.

[9] S. De Agostino. *A Parallel Decoding Algorithm for LZ2 Data Compression,* Parallel Computing, vol. 21, 1995, pp. 1957-1961.

[10] L. Cinque, S. De Agostino and L. Lombardi, *Scalability and Communication in Parallel Low-Complexity Lossless Compression,* Mathematics in Computer Science, vol. 3, 2010, pp. 391-406.

[11] S. T. Klein and Y. Wiseman, *Parallel Lempel-Ziv Coding,* Discrete Applied Mathematics, vol. 146, 2005, pp. 180-191.

[12] S. De Agostino, *LZW Data Compression on Large Scale and Extreme Distributed Systems,* Proceedings Prague Stringology Conference, 2012, pp. 18-27.

[13] J. A. Storer and T. G. Szimansky, *Data Compression via Textual Substitution,* Journal of ACM, vol. 24, 1982, pp. 928-951.

[14] T. A. Welch, *A Technique for High-Performance Data Compression,* IEEE Computer, vol. 17, 1984, pp. 8-19.

[15] J. A. Storer, *Data Compression: Methods and Theory,* Computer Science Press, 1988.

[16] S. De Agostino, *Bounded Size Dictionary Compression: Relaxing the LRU Deletion Heuristic,* International Journal of Foundations of Computer Science, vol. 17, 2006, pp. 1273-1280.

[17] T. C. Bell, J. G. Cleary and I. H. Witten, *Text Compression,* Prentice Hall, 1990.

[18] H. J. Karloff, S. Suri and S. Vassilvitskii, *A Model of Computation for MapReduce,* Proc. SIAM-ACM Symposium on Discrete Algorithms (SODA 10), SIAM Press, 2010, pp. 938-948.

[19] S. De Agostino, *A Robust Approach to Large Size Files Compression using the MapReduce Web Computing Framework,* International Journal on Advances in Internet Technology, vol. 7, 2014, pp. 29-38.

# Locator/Id Split Protocol Improvement for High-Availability Environment

Full-fledged LISP and VRRP simulation modules for OMNeT++

Vladimír Veselý, Ondřej Ryšavý
Department of Information Systems
Faculty of Information Technology, Brno University of Technology (FIT BUT)
Brno, Czech Republic
e-mail: {ivesely, rysavy}@fit.vutbr.cz

*Abstract*—**Locator/Id Split Protocol is a currently discussed alternative to deal with the traditional IP drawbacks (like cumbersome support of device mobility or more importantly default-free zone routing table growth due to the increased demand for multihoming and traffic engineering). This work outlines LISP and its properties for high-availability environments employing first-hop redundancy protocols. This paper also suggests LISP improvement for map-cache synchronization that should impact its routing performance. For this cause, two new simulation models (LISP and VRRP) are introduced that are behaviorally fully RFC compliant.**

***Keywords-LISP; VRRP; map-cache synchronization; OMNeT++***

## I. Introduction

The Automated Network Simulation and Analysis (ANSA) project running at our university is dedicated to developing the variety of simulation models compatible with RFC specifications or referential implementations. Subsequently, these tools allow a formal analysis of real networks and their configurations. They may be publicly used as the routing/switching baseline for further research initiatives, i.e., in simulations for proving (or disproving) certain aspects of technologies and/or related protocols.

**Locator/ID Split Protocol (LISP)** emerged as one of the routing alternatives for Internet Protocol (IP) networks. LISP is the response to problems described in RFC 4984 [1] and RFC 6227 [2]. It should solve Internet architecture problems such as unscalable default-free zone (DFZ) routing, cumbersome mobility and prefix deaggregation caused by multihoming and ingress traffic engineering.

IP address functionality is dual. It serves for identification (“which device is it?”) and localization (“where is the device?”) purposes. The main idea behind LISP is to remove this duality so that there are networks doing routing either based on locators (i.e., transit networks like DFZ) or identifiers (i.e., edge end networks). LISP accomplishes this by splitting the IP addresses into two distinct namespaces: a) **Endpoint Identifier (EID)** namespace (so called LISP site), where each device has unique address; b) **Routing Locator (RLOC)** namespace with addresses intended for localization. There is also a non-LISP namespace where direct LISP communication is (even intentionally) not supported.

Apart from namespaces, there also exist: a) specialized routers (called **tunnel router** a.k.a. **xTR**) spanning between different namespaces; b) dedicated devices maintaining mapping system; and c) proxy routers allowing communication between LISP and non-LISP world.

A LISP mapping system performs lookups to retrieve a set of RLOCs for a given EID. Tunnel routers between namespaces utilize these EID-to-RLOC mappings to perform map-and-encapsulation (see RFC 1955 [3]). The original (inner) header (with EIDs as addresses) is encapsulated by a new (outer) header (with RLOCs as addresses), which is appended when crossing borders from EID to RLOC namespace. Whenever a packet is crossing back from RLOC to EID namespace, the packet is decapsulated by stripping outer header off.

Queries performing EID-to-RLOC mapping are data-driven. This behavior means that a new data transfer between LISP sites may require a mapping lookup, which causes that data dispatch is stopped until mapping is retrieved. This behavior is analogous to the domain-name system (DNS) protocol and allows LISP to operate decentralized database of EID-to-RLOC mappings. Replication of whole (potentially large-scale) database is unnecessary because mappings are accessed on-demand, just like as in DNS a host does not need to know complete domain database. Tunnel routers maintain **map-cache** of recently used mappings to improve performance of the system.

LISP is being successfully deployed in enterprise networks, and one of its most beneficial use-cases is for data-centers networking. An important feature of any data-center is its ability to maintain high-availability of provided services. This goal is accomplished mainly with redundancy. In the case of the outage, service delivery is not affected because of redundant links, devices or power sources. **Virtual Router Redundancy Protocol (VRRP)** is among related protocols and technologies guaranteeing redundancy and helping to achieve high-availability.

VRRP is widely adopted protocol providing redundancy of default-gateway (crucial L3 device that serves as exit/entry point to a given network). VRRP is IETF’s response for

Cisco's proprietary Hot Standby Routing Protocol (HSRP) and Gateway Load Balancing Protocol (GLBP) delivering same goals.

VRRP combines redundant first hop routers into virtual groups. One master router actively forwards clients traffic within each group, where others in the group are backing its functionality. Backup routers are periodically checking liveness of the master waiting ready to substitute it in case of failure. Switching to a new active router is transparent from the host's perspective thus no additional configuration or special software is needed.

This paper introduces two new simulation modules, which create a part of the ANSA project and which extend the functionality of the INET framework in OMNeT++. Subsequently, they are employed as measurement tools supporting proposed LISP map-cache synchronization technique.

This paper has the following structure. The next section covers a quick overview of existing simulation modules. Section III describes the design of relevant LISP and VRRP models. Section IV deals with a map-cache synchronization mechanism – how synchronization works, how it is implemented and how it should aid devices to run LISP and VRRP simultaneously. Section V presents validation scenarios for outlined implementations and shows promising results backing up improvement's impact on LISP operation. The paper is summarized in Section VI together with unveiling of our plans.

## II. State of the Art

This section outlines the current state of the art of available LISP and VRRP implementations for simulator environments.

Limited LISP implementation was created [4] to support LISP MobileNode NAT traversal [5]. However, it is intended for outdated INET-20100323 and OMNeT++ 4.0. Previously, LISP map-cache performance have been evaluated employing high-level simulation that is not taking into account protocol implementation specifics [6].

We are not aware that any VRRP (or another first-hop redundancy protocol) implementation is supported by other major simulators like NS-2/3 or OPNET.

According to our knowledge, OMNeT++ 4.6 (discrete event simulator) and INET 2.4 (framework for wired networks simulation) do not support VRRP simulation modules at all. LISP is supported partially as the result of our previous research effort [7].

Thus, we have implemented LISP and VRRP modules by ourselves in order to have reliable components for subsequent research (i.e., evaluation of proposed improvements).

## III. Implementation

### A. *LISP – Theory of Operation*

LISP is being codified within IETF [8]. The main core and functionality is described in RFCs 6830-6836.

LISP supports both IPv4 and IPv6. Moreover, LISP is agnostic to address family thus it can seamlessly work with any upcoming network protocol. Transition mechanisms are part of the protocol standard. Hence, LISP supports communication with legacy non-LISP world. LISP places additional UDP header succeeded by LISP header between inner and outer header. LISP uses reserved port numbers – 4341 for data traffic and 4342 for signalization.

Basic components of the LISP architecture are **Ingress Tunnel Router (ITR)** and **Egress Tunnel Router (ETR)**. Both are border devices between EID and RLOC space; the only difference is in which direction they operate. The single device could be either ITR-only or ETR-only or ITR and ETR at the same time (thus abbreviation xTR). ITR is the exit point from EID space to RLOC space, which encapsulates the original packet. This process may consist of querying mapping system followed by updating local map-cache, where EID-to-RLOC mapping pairs are stored for a limited time to reduce signalization overhead. ETR is the exit from RLOC space to EID space, which decapsulates packet. Outer header, auxiliary UDP, and LISP headers are stripped off. ETR is responsible for registering all LISP sites (their EID addresses) and by which RLOCs they are accessible.

LISP mapping system consists of two components – **Map Resolver (MR)** and **Map Server (MS)**. The list below contains all LISP control messages responsible for mapping system signalization. They are without inner header – just outer header, followed by UDP header (with source and destination ports set on 4342), followed by appropriate LISP message header.

- *LISP Map-Register* – Each ETR announces LISP site(s) to the MS with this message. Each registration contains authentication data and the list of mappings and their properties.
- *LISP Map-Notify* – UDP cannot guarantee message delivery. MS may optionally (when proper bit is set) confirm reception of *LISP Map-Register* with this message.
- *LISP Map-Request* – ITR generates this request whenever it needs to discover current EID-to-RLOC mapping and sends a message to preconfigured MR.
- *LISP Map-Reply* – This is a solicited response from the mapping system to the previous request and contains all RLOCs to a certain EID together with their attributes.

MR processes ITR's *LISP Map-Requests*. Either MR responds with *LISP Negative Map-Reply* if queried address is from a non-LISP world (not EID), or *LISP Map-Requests* is delegated further into mapping system to appropriate MS.

Every MS maintains **mapping database** of LISP sites that are advertised by *LISP Map-Register* messages. If MS receives *LISP Map-Request* then: a) either MS responds directly to querying ITR; or b) MS forwards request towards designated ETR that is registered to MS for target EID. xTRs perform **RLOC probing** (checking of non-local locator liveness) in order to always use current information.

Each RLOC is accompanied by two attributes – priority and weight. **Priority** (one byte long value in the range from 0 to 255) expresses each RLOC preference. The locator with the lowest priority is preferred for outer header address. Priority value 255 means that the locator must not be used for traffic forwarding. Incoming communication may be load-balanced based on the **weight** value (in the range from 0 to

100) between multiple RLOCs sharing the same priority. Zero weight means that RLOC usage for load-balancing depends on ITR preferences.

### *B. LISP – Design of a Simulation Module*

Simulation model of LISP xTR, MR and MS functionality is currently implemented as `LISPRouting` compound module. It consists of five submodules that are depicted in Figure 1 and described in Table I below. `LISPRouting` exchanges messages with `UDP`, IPv4 `networkLayer`, and IPv6 `networkLayer6` modules. Implementation is fully in compliance with RFC 6830 [9] and RFC 6833 [10].

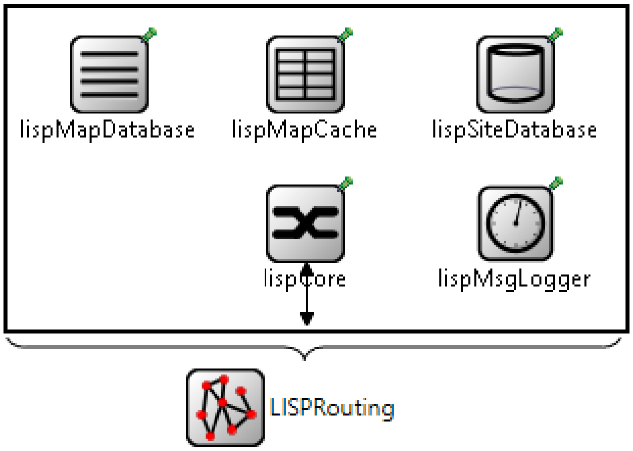


Figure 1. LISPRouting module structure

TABLE I. DESCRIPTION OF LISPROUTING SUBMODULES

| Name | Description |
|---|---|
| `lispCore` | Module handles LISP control and data traffic. It independently combines functionality of ITR, ETR, MR and MS. This involves: encapsulation and decapsulation of data traffic; ETR site registration and MS site maintenance; ITR performing lookups; MR delegating requests. |
| `lispMapDatabase` | Each xTR maintains configuration of its LISP sites (i.e., which RLOCs belong to a given EID or which local interfaces are involved in LISP) that is used by control-plane during registration or for RLOC probing. |
| `lispMapCache` | Local LISP map-cache that is populated on demand by routing data traffic between LISP sites. Each record (EID-to-RLOC mapping) has its separate handling (i.e., expiration, refreshment, availability of RLOCs). |
| `Lisp SiteDatabase` | MS's database that maintains LISP site registrations by ETRs. It contains site-specific information (e.g., shared key, statistics of registrars and most importantly known EID-to-RLOC mappings). |
| `lisp MsgLogger` | This module records and collects statistics about LISP control plane operation, e.g. number, types, timestamps and length of messages. |

### *C. VRRP – Theory of Operation*

VRRP specification is publicly available as RFC standard – RFC 3768 [11] describes IPv4-only VRRPv2 and RFC 5798 [12] describes dual IPv4+IPv6 VRRPv3. VRRPv2 routers send control messages to multicast address 224.0.0.18. VRRPv3 routers use ff02::12 for IPv6 communication. VRRP has its own reserved IP protocol number 112.

Clustered redundant routers form a VRRP group identified by **Virtual Router ID** (**VRID**). Within the group, a single router (called **Master**) is elected based on announced **priority** (a number in the range from 1 to 255). Higher priority means superior willingness to become Master, zero priority causes router to abstain from being Master. In the case of equal priority, binary higher IP address serves as tie-breaker. VRRP election process is always preemptive (unlike to non-preemptive HSRP or GLBP). Peemption means that the router with the highest priority always wins to be the Master no matter whether the group already have other Master elected. Only Master actively forwards traffic. Remaining routers (called **Backup**s) are just listening and checking for Master's keep-alive messages.

Hosts have configured virtual IP address as their default-gateway. Only Master responds to *ARP Requests* for this IP. This IP address has assigned reserved MAC address – 00:00:5e:00:01:$$ for VRRPv2 and 00:00:5e:00:02:$$ for IPv6 (where $$ is VRID). Whenever VRRP group changes to a new Master, *ARP Gratuitous Reply* is generated in order to rewrite association between the interface and reserved MAC in CAM table(s) of switch(es). This allows transparent changing of Masters for hosts during the outage.

VRRP has only one type of control message – *VRRP Advertisement*. If Master is not elected, then, VRRP routers exchange advertisements to determine which one is going to be a new Master. If Master is already elected, then, only Master is sending *VRRP Advertisements* to inform Backups that it is up and correctly running. *VRRP Advertisement* is generated whenever advertisement timer ($AT$) expires (by default every 1 second). If this interval is set to a lower value, then, Master's failure is detected faster but protocol overhead increases. Master down interval ($MDI$) resets with each reception of an advertisement message. Backup, which expires the $MDI$ sooner, becomes a new Master. Value of $MDI$ depends on priority of each VRRP router according to (1). The highest (best) priority Backup times out first (because of the lowest $skew\ time$) and thus takes over role as a new Master before others.

$$MDI = 3 \times AT + \overbrace{\frac{256-priority}{256}}^{skew\ time} \quad (1)$$

### *D. VRRP – Design of Simulation Module*

VRRP version 2 is implemented as `VRRPv2` compound module connected with `networkLayer`. Module is a container for dynamically created instances of `VRRPv2VirtualRouter` during simulation startup. Each instance handles particular VRRP group operation on a given interface. Its structure is depicted in Figure 2, and a brief description of the functionality follows in Table II. Both modules together implement full-fledged VRRPv2 with the same finite-state machine (FSM) as in [11]. VRRP FSM's states *Init*, *Backup* and *Master* reflect VRRP router role and govern control message generation and processing.

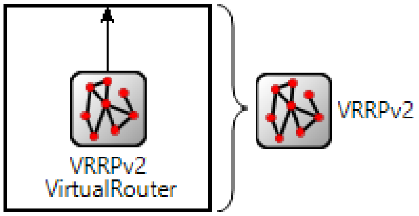


Figure 2. VRRP modules structure

TABLE II. DESCRIPTION OF VRRP MODULES

| Name | Description |
|---|---|
| `VRRPv2` | Responsible for the creation of `VRRPv2VirtualRouters` according to the startup configuration and forwarding VRRP messages to/from them between appropriate gates. |
| `VRRPv2 VirtualRouter` | This module governs *VRRP Advertisements* processing, transition between states and directs ARP for a single VRRP group. |

## IV. CONTRIBUTION

Assume multiple redundant routers are acting as first hops in high-availability scenario. Those routers are simultaneously clustered into VRRP groups and act as LISP's xTRs – they run LISP and VRRP at the same time.

The performance of map-and-encap depends on the fact whether xTR's map-cache contains valid EID-to-RLOC mapping or not. Dispatched data traffic drives Map-cache record creation. If map-cache misses the mapping, then, a mapping system needs to be asked and initiating data traffic is meantime dropped. Packet dropping is a valid step as long as the mapping is not discovered because map-and-encap cannot occur without proper information. The rationale behind this behavior is the same as in the case of ARP throttling [13], where any triggering traffic should be discarded to protect control-plane processing and prevent superfluously recurrent mapping system queries.

Each xTR has its own map-cache and its content may differ even within the same LISP site because each cache record may be initialized by different traffic. Hence, xTRs can easily experience severe packet drops and LISP control message storms due to the map-cache misses when Master change occurs within VRRP group.

This problem is described as the one of LISP weak-points in [14] and theoretically investigated in [15]. The viable solution would be to provide map-cache content synchronization that should minimize map-cache misses upon failure. Inspired by that, we present our solution addressing this problem.

We have decided to implement it as a technique maintaining synchronized map-caches within a predefined **synchronization set (SS)** of ITRs. Any solicited *LISP Map-Reply* triggers synchronization process among SS members.

Each record in the map-cache is equipped with a time-to-live (TTL) parameter. TTL expresses for how long the record is considered to be valid and usable for map-and-encap. Map-caches within SS must maintain the same TTL on shared records; otherwise a loss of synchronization might occur (on some ITRs, identical records could expire because of no demand by traffic).

We have implemented two modes of synchronization:

1) *Naïve* – The whole content of map-cache is transferred to SS. All mappings are then updated according to the new content and TTLs are reset. This approach works fine, but it obviously introduces significant transfer overheads;
2) *Smart* – Only record that caused synchronization is transferred. Moreover, we bound this mode with following policy. When TTL expires, the ITR must check record usage during the last minute (one minute should be a period enough long to detect ongoing communication). If the mapping has not been used, then, it is removed from the cache. Otherwise, its state is refreshed by query followed by synchronization.

Synchronization itself is done with the help of two new LISP messages:

- *LISP CacheSync* – Message contains map-cache records that are being synchronized and authentication data protecting SS members from spoofed messages;
- *LISP CacheSync Ack* – Because LISP leverages UDP, it cannot guarantee message delivery. However, we decided to employ the same principle as for *LISP Map-Register* and *LISP Map-Notify*. Hence, *LISP CacheSync* delivery may be optionally confirmed by echoing back *LISP CacheSync Ack* message.

This approach guarantees that devices within SS could forward rerouted LISP data traffic without packet loss or interruption because they share the same content as ITR's map-cache of malfunctioned former VRRP Master.

## V. TESTING

In this section, we provide information regarding validation of LISP and VRRP simulation models. This is necessary in order to build up reliability of used tools for subsequent evaluation of proposed map-cache synchronization technique.

We have built exactly the same real network topologies as for simulations. We captured and analyzed (using transparent switchport analyzers and packet sniffers) relevant messages exchanged between devices for both LISP and VRRP functionality validation. We compared the results with the behavior of a referential implementation running on Cisco routers (namely C7200 with IOS version c7200-adventerprisek9-mz.152-4.M2) and host stations.

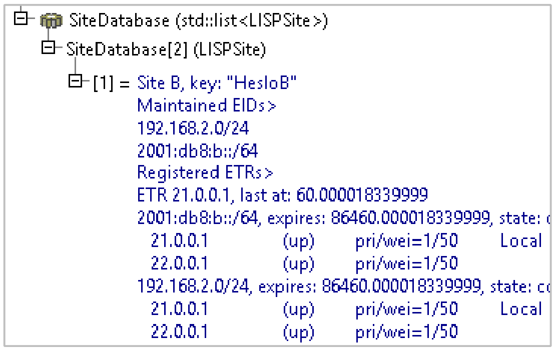
### A. LISP Functionality

We have verified LISP implementation on the topology depicted in Figure 6. Simulation network contains two sites – green areas "Site A" (interconnected by switch `S1`, bordered by `xTR_A1` and `xTR_A2`) and "Site B" (interconnected by `S2`, bordered by `xTR_B1` and `xTR_B2`). The topology contains router `MRMS`, which acts as MR and MS for both sites. IPv4 only capable core (red area) is simulated by a single `Core` router. Static routing is employed to achieve mutual connectivity across core. `HostA` and `HostB` are dual-stack devices, where `HostA` is scheduled to ping `HostB` after second successful site registration (at `t=70s`). `MRMS` is allowed to proxy-reply on mapping requests for "Site A". All RLOCs are configured with priority 1 and weight 50 to achieve equal load balancing for incoming traffic.

Testing scenario beginning is aligned with initialization of `xTR_A1`'s LISP process that freshly starts after the reboot. The list of important phases is briefly described below:

#1) First of all, each ETR starts RLOC probing, which is polling mechanism that checks reachability of announced locators. Each ETR sends *LISP Map-Request* with probe-bit set on to queried RLOC address (e.g., `xTR_A1` is probing `xTR_A2`'s locator 12.0.0.1). Neighboring `xTR_*` then responds with *LISP Map-Reply* with probe-bit set announcing state of its RLOC interface. This process repeats by default every minute. The lower RLOC-probe timer is, the sooner RLOC outage is detected but protocol's overhead increases. Also Cisco's LISP implementation queries same RLOC for each assigned EID.

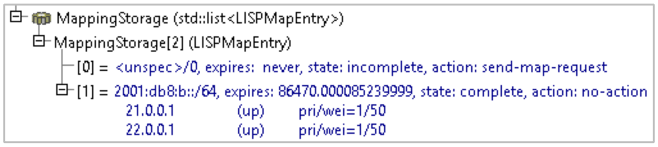
#2) ETRs sends registration about their EID sites towards MS. Each `xTR_*` generates *LISP Map-Register* message. Registration process repeats every 60 seconds in order to keep mappings up-to-date. *LISP Map-Register* contains all EID-to-RLOC mapping properties (i.e., EID, TTL, RLOC statuses, and attributes). For phase #2 illustration, figure 3 shows `xTR_B1`'s "Site B" registration after #1.



```
⊟ SiteDatabase (std::list<LISPSite>)
  ⊟ SiteDatabase[2] (LISPSite)
    ⊟ [1] = Site B, key: "HesloB"
            Maintained EIDs>
            192.168.2.0/24
            2001:db8:b::/64
            Registered ETRs>
            ETR 21.0.0.1, last at: 60.000018339999
            2001:db8:b::/64, expires: 86460.000018339999, state: c
              21.0.0.1        (up)      pri/wei=1/50      Local
              22.0.0.1        (up)      pri/wei=1/50
            192.168.2.0/24, expires: 86460.000018339999, state: co
              21.0.0.1        (up)      pri/wei=1/50      Local
              22.0.0.1        (up)      pri/wei=1/50
```

Figure 3. xTR_B1's registration of "Site B"

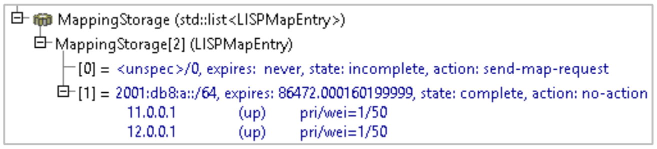
#3) `HostA` initiates ping to `HostB`'s address 2001:db8:b::99. *ICMP Echo Request* is delivered to `xTR_A1` (hosts default-gateway), where it triggers LISP query because that particular EID-to-RLOC mapping is currently unknown. First ping is dropped due to that. `xTR_A1` sends *LISP Map-Request* to MS. MRMS performs lookup on its site database and delegates request to one of the designated ETRs, in this case, `xTR_B1`. `xTR_B1` responds with *LISP Map-Reply* with current mapping (to EID 192.168.2.0/24 belongs two RLOCs 21.0.0.1 and 22.0.0.1). Figure 4 illustrates this result.



```
⊟ MappingStorage (std::list<LISPMapEntry>)
  ⊟ MappingStorage[2] (LISPMapEntry)
     ├ [0] = <unspec>/0, expires:  never, state: incomplete, action: send-map-request
     ⊟ [1] = 2001:db8:b::/64, expires: 86470.000085239999, state: complete, action: no-action
              21.0.0.1        (up)      pri/wei=1/50
              22.0.0.1        (up)      pri/wei=1/50
```

Figure 4. Content of xTR_A1's map-cache after phase #3

#4) Second ping arrives on `xTR1_A1`. Because mapping is known, it is encapsulated with outer header as LISP carrying data (marked *LISP Data* message) and sent to one of `xTR_B*` after random selection of equally preferred locators. In our case, *LISP Data* is delivered to `xTR_B2` where original ping is decapsulated and forwarded further to end destination. `HostB` responds with *ICMP Echo Reply* that is passed to its default-gateway (`xTR_B1`). Over here the same process as in #4 repeats – ping is dropped and mapping query triggered. Only this time, MS replies directly to *LISP Map-Request*. `MRMS` is allowed to send *LISP Map-Reply* instead of designated ETR because of proxy-reply for "Site A". Figure 5 shows the result.



```
⊟ MappingStorage (std::list<LISPMapEntry>)
  ⊟ MappingStorage[2] (LISPMapEntry)
     ├ [0] = <unspec>/0, expires:  never, state: incomplete, action: send-map-request
     ⊟ [1] = 2001:db8:a::/64, expires: 86472.000160199999, state: complete, action: no-action
              11.0.0.1        (up)      pri/wei=1/50
              12.0.0.1        (up)      pri/wei=1/50
```

Figure 5. Content of xTR_B1's map-cache after phase #4

#5) Third and other consecutive pings pass without experiencing any drop because both default-gateways have proper EID-to-RLOC mappings.

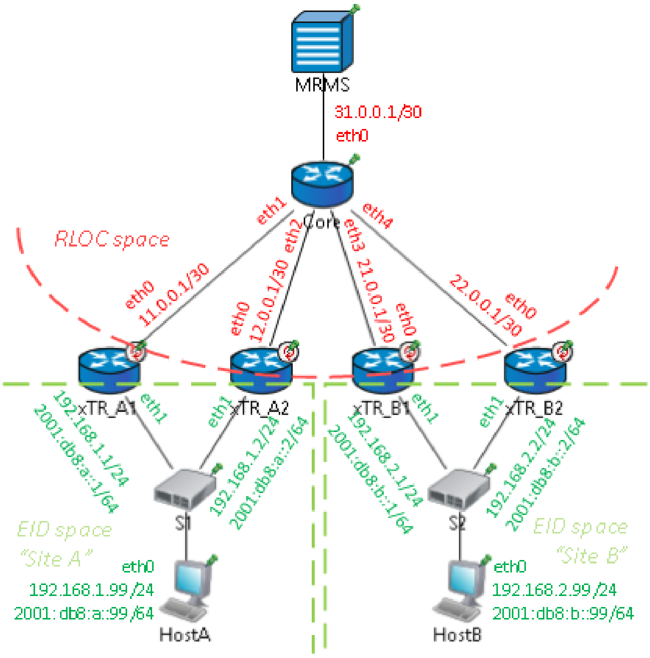


Figure 6. LISP testing topology

Phases of LISP operation are compared to simulation and real network in Table III. For clarity and due to limited space, only some messages are recorded for #1, #2 and #3. Nevertheless, omitted messages do not show significant deviations.

TABLE III. TIMESTAMP COMPARISON OF LISP MESSAGES

| Phase | Message | Sender | Simul. [s] | Real [s] |
|---|---|---|---|---|
| #1 | *LISP Map-Req. Probe* | `xTR_A1` | 0.000 | 0.000 |
| | *LISP Map-Rep. Probe* | `xTR_A2` | 0.000 | 0.063 |
| #2 | *LISP Map-Register* | `xTR_A1` | 60.000 | 60.567 |
| #3 | *ICMP Echo Request* | `HostA` | 70.000 | 70.000 |
| | *LISP Map-Request* | `xTR_A1` | 70.000 | 70.361 |
| | *LISP Map-Reply* | `xTR_B1` | 70.000 | 70.460 |
| #4 | *ICMP Echo Request* | `HostA` | 72.000 | 71.931 |
| | *LISP Data* | `xTR_A1` | 72.000 | 71.944 |
| | *ICMP Echo Reply* | `HostB` | 72.000 | 71.962 |
| | *LISP Map-Request* | `xTR_B1` | 72.001 | 72.852 |
| | *LISP Map-Reply* | `MRMS` | 72.001 | 72.889 |
| #5 | *ICMP Echo Request* | `HostA` | 74.000 | 74.011 |
| | *ICMP Echo Reply* | `HostB` | 74.001 | 74.177 |

## B. *VRRP Functionality*

We have verified VRRP functionality on the topology depicted in Figure 7. Simulation network contains two VRRP routers (`GW1` and `GW2`) clustered in VRID 10, one switch (`SW`) interconnecting devices on local segment, one host (`Host`) and one router (`ISP`) substituting communication outside LAN. Both VRRP routers are configured with the default priority, default *AT* value and virtual default-gateway IP address set to 192.168.10.254.

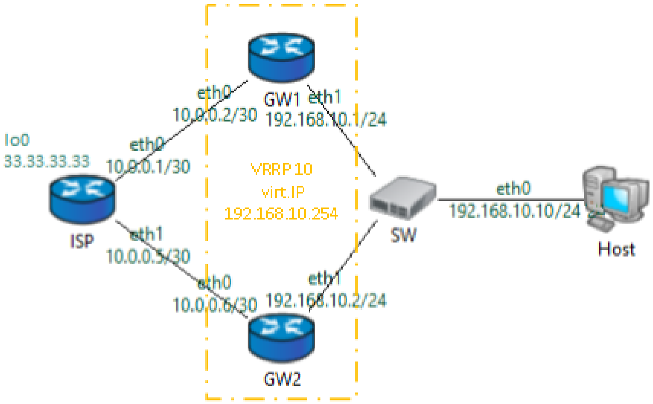


Figure 7. VRRP testing topology

For this test, we scheduled that original Master (`GW2`) would go down (at `t=20s`) and back up (at `t=30s`). Meantime, `Host` starts pinging (at `t=10s`) Internet address 33.33.33.33 every second where traffic goes via virtual default-gateway. Scenario beginning (phase #1 at `t=0s`) is aligned with initialization of VRRP process.

Test goes through following phases:

#1) Both `GW1` and `GW2` immediately transit from *Init* state to *Backup* and are waiting to hear *VRRP Advertisement* from potential Master.

#2) They both expire *MDI* at the same time (`t=3.609275`, equation (1) yields the same result) and transit to *Master* state. This allows them to send their own *VRRP Advertisement* and discover each other. They compare announced properties in advertisement with their own VRRP settings. `GW2` becomes a new Master. Despite having same priority (value 100), `GW2` address 192.168.10.2 is higher.

#3) If `Host` wants to ping 33.33.33.33, then, the traffic needs to go via default-gateway and `Host` requests IP-to-MAC mapping with the help of *ARP Request*. Message is delivered to `GW1` and `GW2`, but only `GW2` responds with *ARP Reply* because it is Mater. Subsequently, endless ping passes through `GW2`.

#4) `GW2` failure occurs and `GW1` seizes to receive *VRRP Advertisements*. `GW1`'s *MDI* expires and next `GW1` becomes a new Master sending its own *VRRP Advertisements*. But before that, `GW1` sends *ARP Gratuitous Reply* in order to change CAM of `SW`. Meantime, pings are being dropped since moment of failure until `GW1` is elected.

#5) Pings pass through `SW` towards `GW1` and `ISP`.

#6) `GW2` goes up and transits after *MDI* from *Init* to *Backup*. Then, `GW2` transits from *Backup* to *Master* state. `GW2` sends its own *VRRP Advertisement*, which is superior to ones from `GW1`, and *ARP Gratuitous Reply* for virtual default-gateway 192.168.10.254. Immediately when `GW1` hears `GW2`'s advertisement, `GW1` abdicates for being Master router and transits to *Backup* state.

The comparison between timestamps and message confluence can be observed in Table IV.

TABLE IV. TIMESTAMP COMPARISON OF VRRP MESSAGES

| Phase | Message | Sender | Simul. [s] | Real [s] |
|---|---|---|---|---|
| #2 | *VRRP Advertisement* | `GW1` | 3.609 | 3.612 |
| | *VRRP Advertisement* | `GW2` | 3.609 | 4.367 |
| | *VRRP Advertisement* | `GW2` | 4.609 | 5.286 |
| #3 | *ARP Request* | `Host` | 10.000 | 10.000 |
| | *ARP Reply* | `GW2` | 10.000 | 10.034 |
| | *IMCP Echo Request* | `Host` | 10.000 | 10.986 |
| #4 | *VRRP Advertisement* | `GW1` | 23.219 | 23.655 |
| | *ARP Gratuitous Reply* | `GW1` | 23.219 | 23.643 |
| #6 | *VRRP Advertisement* | `GW2` | 33.718 | 33.612 |
| | *ARP Gratuitous Reply* | `GW2` | 33.718 | 33.611 |

Please notice that Cisco's VRRP implementation sends two *ARP Gratuitous Replies* before any VRRP advertisement. After we had observed this, we implemented another FSM in our VRRP module to accommodate this behavior. However, the routing outcome from `Host` perspective is same no matter on chosen FSM.

## C. *Map-Cache Synchronization*

The goal of the following test is to show the impact of our synchronization technique on a packet drop rate and a number of map-cache misses. A scenario is focused on cache misses due to the missing mapping rather than expired ones because of default TTL (1 day). Five minutes time slot with the single

VRRP Master outage is the simplest illustration of how to compare the impact of map-cache synchronization.

We prepared simulation topology that contains a single LISP site with two routers (`xTR1` and `xTR2`), which provide highly-available VRRP default-gateway for two hosts interconnected by switch `SW`. `Host1` and `Host2` are pinging IPv4 EIDs randomly thus generating traffic that triggers LISP mapping system queries. All routing is done statically. Hence, there is no need to employ routing protocol on `Core` router. We prepared special xTR called `xTR_Responder` that: a) registers destination EIDs; and b) responds to hosts ICMP messages. The whole topology is depicted in Figure 8.

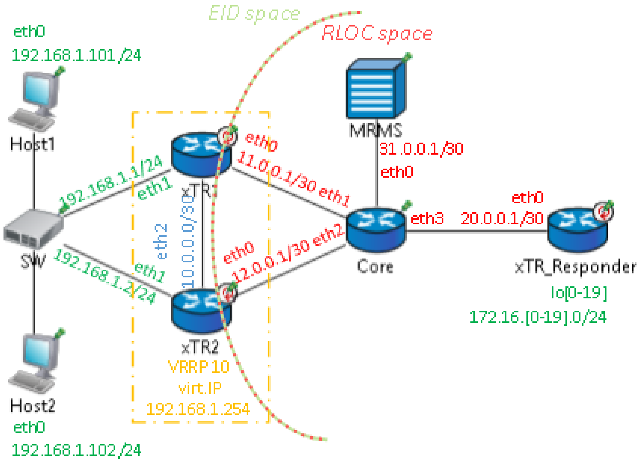


Figure 8. LISP map-cache synchronization testing topology

We scheduled following phases for the test run:

#1) At first, all xTRs register their EIDs. In the case of `xTR_Responder`, EID space is modeled with the help of loopback interfaces – twenty of them ranging with addresses from 172.16.0.0/24 to 172.16.19.0/24 reachable via single RLOC 20.0.0.1. In case of `xTR1` and `xTR2`, EID 192.168.1.0/24 is reachable via two RLOCs 11.0.0.1 and 12.0.0.1.

#2) `xTR1` and `xTR2` form VRRP group with VID 10 and virtual address 192.168.1.254, which is used by `Host1` and `Host2` as default-gateway. `xTR1` is Master because of higher priority (`xTR1` has 150, `xTR2` only 100) as long as it is operational.

#3) `Host1` starts pinging ten random EIDs in range from 172.16.0.0/24 to 172.16.9.0/24. Because EIDs are chosen randomly, they may be duplicate. Each first ICMP packet causes mapping query and is dropped.

#4) Then, `xTR1` failure occurs right before a new LISP registration (at `t=119s`). Hosts traffic is diverted to a new VRRP Master, which is `xTR2`.

#5) After #4, also `Host2` starts to ping ten random EIDs from 172.16.10.0/24 to 172.16.19.0/24. Same duplicity rule as in #3 applies.

#6) `xTR1` recovers from outage at `t=235s` and once again all hosts traffic goes through it.

Depending on map-cache synchronization type, additional map-cache misses might occur. `xTR1` and `xTR2` synchronized themselves via 10.0.0.0/30 connection, which forms dedicated SS. `xTR1` uses address 10.0.0.1 and `xTR2` address 10.0.0.2.

The scenario has been tested with three simulation configurations each representing different map-cache synchronization technique: α) no synchronization at all (default LISP behavior); β) naïve mode; and γ) smart mode. Impact on map-cache is summarized in Table V for all previously mentioned different configuration runs.

Before interpreting the results, please note that `Host1` randomly (using same seeds for all three runs) chose 8 different EIDs, `Host2` 6 EIDs, totally 14 distinct ping destinations.

TABLE V. MAP-CACHE MISSES FOR DIFFERENT CONFIGURATIONS

| **Phase** | **α cache misses** | | **β cache misses** | | **γ cache misses** | |
|---|---|---|---|---|---|---|
| | *xTR1* | *xTR2* | *xTR1* | *xTR2* | *xTR1* | *xTR2* |
| #3 | 8 | 0 | 8 | 0 | 8 | 0 |
| #5 | 0 | 14 | 0 | 6 | 0 | 6 |
| #6 | 14 | 0 | 0 | 0 | 0 | 0 |
| **Total** | 22 | 14 | 8 | 6 | 8 | 6 |

Without any synchronization, traffic diversion to a new VRRP Master always causes misses due to unknown mappings. We can see this in phases #5 and #6 for α-run, when router starts to dispatch LISP data with the empty map-cache.

If the synchronization is employed, then, only new destinations lead to map-cache miss. The reason is that a new VRRP Master already have mappings discovered by neighbor xTR. Hence, there is a difference in phase #5 for α-run (empty cache) and β/γ-runs (cache in sync with SS member). β-and γ-runs are equal in the number of cache misses but γ- run is more effective in protocol overhead. The difference (36 cache misses versus 14) would be even more significant in the case of multiple VRRP Master outages. Please note that every map-cache miss is also connected with the data packet drop.

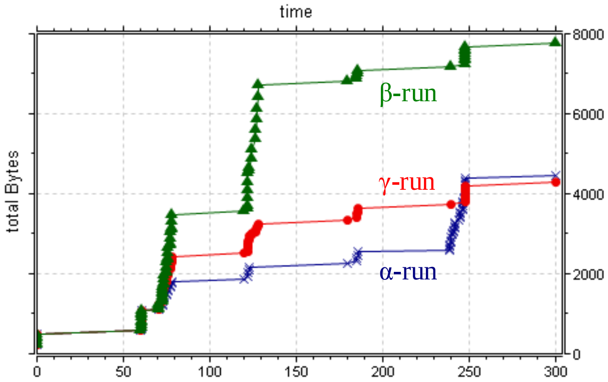


Figure 9. Total size of processed LISP control messages by xTR1

In order to compare synchronization modes, we conducted measurement taking into account all LISP control messages processed by `lispCore` module, namely their packet sizes. We assume that larger size is always greater burden for router's control plane processing. Figure 9 shows the results (α-run = blue crosses, β-run = green triangles, γ-run = red circles), where is visible that smart outperforms naïve. The reason for being less intensive is that only single mapping is transferred during synchronization, not a whole map-cache. Moreover, smart mode is even better than no synchronization because it decreases number of mapping queries. It is even more apparent in the same scenario but with more VRRP outages (see [16]).

## VI. Conclusion

In this paper, we presented a detailed description of LISP and VRRP technologies. We proposed LISP improvement aimed to achieve a better routing performance primarily in high-availability use-cases, e.g. data-centers with mission-critical applications sensitive to packet drops. We evaluated the impact of our improvement using OMNeT++ simulator. In order to achieve this objective and relevant results, we first thoroughly developed LISP and VRRP simulation modules that mimic behavior of real implementations.

Validation testing against a real-life topology shows very reasonable time variations for LISP and VRRP functionality. However, simulation results are affected by simpler simulated control-plane and the simulated processing time does not include all processes running on a real router. Hence, some simulation timestamps in Table III and IV are below one millisecond accuracy. Time variation observable on real Cisco devices is caused by three factors: a) control-plane processing delay and internal optimizations; b) packet pacing avoiding race conditions; and c) inaccuracy in timing of certain events in real-life network. Nevertheless, the routing outcomes of simulated and real network are exactly same when taking into account accuracy in order of seconds.

During our tests, we closely observed RLOC-probing algorithm that Cisco devices are using to verify locator reachability. ITR is checking assigned locators for each configured EID. Although this often leads to repeated check of the same locator multiple times, which represents scalability issue in larger networks. Therefore, we already implemented enhancement that reduces LISP protocol overhead and its precise evaluation is a future research task.

We plan to carry on the work on simulation modules to further test them in more realistic network simulations. Proxy ITR/ETR capability, solicit map-requests and different mapping distribution systems (e.g., LISP-ALT, LISP-DDT) are on our LISP development roadmap. We would like to upgrade VRRP to support IPv6 addresses and all features of VRRP version 3.

All source codes could be downloaded from GitHub repository [16]. Real packet captures and simulation datasets for the results reproduction could be downloaded from Wiki of the repository mentioned above. More information about ANSA project is available on its homepage [17].

## Acknowledgment

The VRRP simulation core is work by Petr Vítek, a former student of the Brno University of Technology. This work was supported by following organizations and research grants:

- FIT-S-14-2299 supported by Brno University of Technology;
- IT4Innovation ED1.1.00/02.0070 supported by Czech Ministry of Education Youth and Sports.

## References

[1] D. Meyer, L. Zhang, and K. Fall, "RFC 4984: Report from the IAB Workshop on Routing and Addressing," September 2007. [Online]. Available: http://tools.ietf.org/html/rfc4984.

[2] T. Li, "RFC 6227: Design Goals for Scalable Internet Routing," May 2011. [Online]. Available from: http://tools.ietf.org/html/rfc6227.

[3] R. Hinden, "RFC 1955: New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG," June 1996. [Online]. Available: http://tools.ietf.org/html/1955.

[4] D. Klein, M. Hoefling, M. Hartmann, and M. Menth, "Integration of LISP and LISP-MN into INET," in Proceedings of the IEEE 5th International ICST Conference on Simulation Tools and Techniques, Desenzano del Garda, Italia, March 2012, pp. 299-306, ISBN 978-1-4503-2464-9.

[5] D. Klein, M. Hartmann, and M. Menth, "NAT Traversal for LISP Mobile Node," July 2010. [Online]. Available: http://tools.ietf.org/html/draft-klein-lisp-mn-nat-traversal.

[6] J. Kim, L. Iannone, and A. Feldmann, "A deep dive into the LISP cache and what ISPs should know about it," NETWORKING 2011, May 2011, vol. 6640, pp. 367-378.

[7] V. Veselý, M. Marek, O. Ryšavý, and M. Švéda, "Multicast, TRILL and LISP Extensions for INET," International Journal On Advances in Networks and Services, 2015, vol. 7, no. 3&4, unpublished.

[8] IETF, "Locator/ID Separation Protocol (lisp)," January 2015. [Online]. Available: http://datatracker.ietf.org/wg/lisp/charter/. [Retrieved: January 2015].

[9] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "RFC 6830: The Locator/ID Separation Protocol (LISP)," January 2013. [Online]. Available: http://tools.ietf.org/html/rfc6830.

[10] V. Fuller, "RFC 6833: Locator/ID Separation Protocol (LISP) Map-Server Interface," January 2013. [Online]. Available: https://tools.ietf.org/html/rfc6833.

[11] R. Hinden, "RFC 3768: Virtual Router Redundancy Protocol (VRRP)," April 2004. [Online]. Available: https://tools.ietf.org/html/rfc3768.

[12] S. Nadas, "RFC 5798: Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6," March 2010. [Online]. Available: https://tools.ietf.org/html/rfc5798.

[13] R. Froom, E. Frahim, and B. Sivasubramanian, CCNP Self-Study: Understanding and Configuring Multilayer Switching, Cisco Press, 2005.

[14] D. Saucez, O. Bonaventure, L. Iannone, and C. Filsfils, "LISP ITR Graceful Restart," December 2013. [Online]. Available: https://tools.ietf.org/html/draft-saucez-lisp-itr-graceful-03.

[15] D. Saucez, J. Kim, L. Iannone, O. Bonaventure, and C. Filsfils, "A Local Approach to Fast Failure Recovery of LISP Ingress Tunnel Routers," NETWORKING 2012, May 2012, vol. 7289, pp. 397-408, ISBN: 978-3-642-30044-8.

[16] GitHub, December 2013. [Online]. Available: https://github.com/kvetak/ANSA/wiki/ICNS-2015. [Retrieved: January 2014].

[17] Brno University of Technology, January 2014. [Online]. Available: http://nes.fit.vutbr.cz/ansa/pmwiki.php. [Retrieved: January 2014].

# An Integrated Testbed Environment for the Web of Things

Mina Younan
Computer Science Department
FCI - Minia University
Minia, Egypt
E-mail: mina.younan@mu.edu.eg

Sherif Khattab, Reem Bahgat
Computer Science Department
FCI - Cairo University
Cairo, Egypt
E-mails: {s.khattab, r.bahgat}@fci-cu.edu.eg

***Abstract* – This paper proposes a testbed architecture for the Web of Things (WoT) using simple components. Sensor networks have become one of the most researched topics currently, due to proliferation of devices equipped with sensors and actuators for monitoring and controlling their surrounding environment (e.g., places and devices). Although simulators, like Cooja, and Web sites, like Thingspeak, give the ability to build simple Internet of Things (IoT) and WoT applications, they are not compatible with many testing purposes in WoT. Getting real datasets that cover the main features of WoT is one of the most important factors in WoT testing and research. The proposed testbed environment allows for generating datasets and using them offline and online. It integrates small equipment elements (sensors and actuators) that convert things into Smart Things (STHs). Moreover, it augments IoT by STHs virtualization through Web applications. The main components and detailed design of the testbed are described. Then, a case study of searching for STHs and Entities of Interest (EoIs) is explained using a real dataset generated from the proposed testbed. The proposed testbed architecture incorporates the sense of smart environments and, hence, is expected to enhance testing results in WoT.**

***Keywords – Internet of Things (IoT); Web of Things (WoT); searching in WoT; Smart Things; Test Environment*.**

## I. INTRODUCTION

The number of devices and things connected to the Internet will be increasing and is expected to reach the order of billions by 2020 [1][2], as soon as the Internet Protocol (IP) becomes the core standard in the fields of embedded devices. As a result, the number of Internet users will be less than the number of devices connected to it. The Internet of Things (IoT) focuses on the infrastructure layer needed for connecting things and devices to the Internet. IoT addresses the connectivity challenge by using IP and IPv6 for embedded devices (i.e., 6LoWPAN) [3]. Sensor networks have become one of the most researched topics currently [2] [4]. This is due to the proliferation of devices equipped with sensors and actuators that provide information about and control of their surrounding environments. Sensors allow the state of things (e.g., places, devices, etc.,) that sensors represent to be inferred. In a sense, sensors and actuators convert things to Smart Things (STHs) and things' environments to smart spaces.

The Web of Things (WoT) virtualizes the IoT and focuses on the application layer needed for building useful applications over the IoT. Services, such as searching for STHs and Entities of Interest (EoIs) in the WoT, in addition to Web-based applications for controlling and monitoring services in smart spaces using friendly user interfaces are core power features in the WoT. However, there is no general method for testing and benchmarking research in IoT and WoT [4][5][6][7].

Muhammad et al. [6] summarize differences between concepts of emulators, simulators, and physical testbeds. They concluded that physical testbeds provide more accurate results. MoteLab [4] is a testbed for Wireless Sensor Networks (WSNs). It addresses challenges related to sensors' deployment and the time consumed for building a WSN. It features a Web application to be accessed remotely. The need for WSN testbeds is highlighted by challenges and research topics, which shed light on a specific set of features to be embedded within the testbed and its tools [6][7]. For instance, not only datasets about sensor readings are needed but integrating the readings with information about the underlying infrastructure (i.e., the IoT layer) is needed as well; this integration is the goal of the testbed proposed in this paper.

This paper proposes a testbed architecture for WoT. It addresses the general needs of WoT testing and focuses on the Web search problem and its related issues, such as crawling (i.e., preparing WoT pages for crawling). The problem of how to find STHs and EoIs that have dynamic state that change according to environment events [8][9] has sheer importance in drawing conclusions, deductions, and analysis in various fields. The proposed testbed can be used as a WoT application, which monitors real devices in real-time and can be used as a WoT simulator to do the same process on WoT datasets instead of devices. It aims at collecting datasets that contain information about things (i.e., properties and readings) formatted using multiple markup languages. The collected datasets are designed to help in testing in many problem domains [8][10].

The remainder of the paper is organized as follows. The next section defines dataset requirements. In Section 3, the related work of creating searchable IoT and WoT domains using IoT and WoT simulators and datasets is discussed. Section 4 describes the proposed system architecture. In

Section 5, the implementation of the proposed system is described followed by a case study. Finally, conclusions and important ideas for future work are presented in Section 6.

## II. Dataset Requirements

Things, SThs, resources, and EoIs are main concepts in IoT and WoT. They have differences in meaning but the main goal is that they are used for integrating the physical world into the virtual world [11]. In WoT, what needs to be retrieved (i.e., searched for) includes SThs (e.g., TV sets), EoIs (e.g., buildings), IP addresses, current values [8], and general information like device's web banners [9]. Generally, searching is done on SThs and EoIs that have dynamic locality [8][12][13] caused and fired by other events or objects in the network. For testing and evaluating the search process in IoT and WoT, simulators should reflect as many IoT and WoT challenges to achieve accurate results. To achieve this, datasets are used and replayed by applications that act as emulators of WoT.

In general, the main challenges that face testing of smart spaces, IoT, and WoT are: (1) the huge number of sensors and SThs, which makes communication services, monitoring, and analysis of sensor information require non-trivial amounts of CPU time and storage, (2) the dynamic state of SThs, which means that sensor readings and information about SThs properties and devices to which they are attached are in continuous change, and (3) the non-standardized naming of SThs properties (e.g., name, services, and location) and formats (e.g., microformats and microdata) that are used in WoT applications, which makes retrieval of information about objects and their attached sensors difficult.

To identify dataset requirements for testing the Web search process in WoT, WoT data are classified according to type, static or dynamic. The first type is **static information (IoT level)**, which includes (1) information about *sensors* (e.g., information about sensor properties like ID, name, brand, image, description, authoritative URL, manufacturer, and list of services that it offers) and (2) information about *entities* (e.g., device, thing, and place) including entity properties, such as logical paths, list of hosted devices, and possible states by which the entity is described. The second WoT data type is **dynamic information (WoT level)**, which includes (1) sensor readings or state, (2) current entity state, which changes according to sensor reading or other factors that the entity state depends on [8][14]. A dataset for testing Web search in WoT should ideally contain the following items: (1) files that contain schematics of the buildings and locations of sensors, (2) files that contain other static information about sensors (written in different formats), (3) a file for each sensor type that contains a table for readings of all sensors that have that type as a time series to aid in testing sensor similarity search and analysis [15], and a file for all devices in the network that contains sensor readings as a time series so that it can be used for browsing the WoT. Examples of these files will be described later. For accessing these files, headers of tables (sensor definitions) should follow a certain structure for creation and accessing (e.g., sensor name and virtual and physical location).

## III. Related Work

In the light of the previous requirements, this section discusses the usage of sensor datasets in the literature. To summarize our observations, if the research is only interested in values measured by sensors or in states of EoIs (e.g., being online or offline), then the used dataset is based on the WoT level, whereas if the research is interested in the sensor network infrastructure, then the used dataset is based on the IoT level. An integrated dataset contains information about both sensor readings and network infrastructure, that is, it is based on both IoT and WoT levels.

### *A. IoT Simulations*

There is no general way for simulating IoT [5][6][16]. Moreover, there are situations in which simulators and real datasets containing raw information (e.g., sensor readings [17]) or information about the IoT layer are not enough for modeling an environment under testing, as the datasets miss the sense of one or more of the challenges mentioned earlier and thus, miss the main factors for accurate WoT evaluation. Also, many datasets are not actually related to the problem under investigation, but were generated for testing and evaluating different algorithms or methods in other researches. For instance, an evaluation of WSNs' simulators according to a different set of criteria, such as Graphical User Interface (GUI) support, simulator platform, and available models and protocols, concludes that there is no general way for simulating WSNs, and hence IoT and WoT [5][16]. None of these criteria address the previous challenges. So, it is desirable to embed the unique IoT and WoT challenges within datasets and to make simulators support as much of these challenges.

**WSN Simulators**. Several studies [5][6][16] summarize the differences between existing simulators according to a set of criteria. The Cooja simulator is one of the most valuable tools [5][16] in WSNs that aids researchers to simulate WSNs relatively easily using a supported GUI. Cooja allows to add different types of sensors (motes) and to attach them to binaries or source codes that have been previously developed. Cooja supports applications (e.g., written in the nesC [18] language after building in the TinyOS [19]) for different sensor targets. For example, the RESTful client server application [20] simulates a simple IoT. Cooja has a main advantage that it allows users to create their network using a non-trivially large number of sensors with different types, to get information about sensors (readings and properties), to control sensors, and to change their states as well. However, there are limitations and difficulties for testing the extensible discovery service [10] and sensor similarity search [15] in Cooja, because there is no information about network infrastructure and entities, in particular static information about sensors, written in different formats, and schematics information of the buildings and locations of sensors.

**WSN Physical Testbeds.** Physical testbeds produce accurate research results [6]. Different testbeds are found in this field due to different technologies and network scales. Providing a Web interface for users is a main feature in testbeds. MoteLab [4] supports two ways for accessing the WSNs, (1) offline, by retrieving stored information form a database server and (2) online, by direct access to the physical nodes deployed in the environment under test. Datasets can be downloaded from MoteLab's Web site. However, the WoT challenges mentioned previously are not fully supported in MoteLab. User accessibility in MoteLab is similar to what is done in the proposed testbed.

SmartCampus [21] tackles gaps of experimentation realism, supporting heterogeneity (devices), and user involvement [7] in IoT testbeds. CookiLab [22] is another WSN testbed. It gives users (researchers) the ability to access real sensors deployed in Harvard University. However, it does not support main WoT features, such as sensor formats and logical paths as a property for sensor nodes and entities.

Nam et al. [23] present an Arduino [24] based smart gateway architecture for building IoT. Their architecture is similar to the architecture of the testbed environment proposed in this paper. For example, both their approach and ours use periodic sensor reporting. The Sense Everything, Control Everything (SECE) server stores information sent by Arduinos to be accessible anywhere at any time. Also, they provide the 'Bonjour' application that discovers all connected Arduinos and lists the devices connected on each Arduino. The information is sent in JavaScript Object Notation (JSON) format back to the application [23]. However, the framework does not cover all scenarios that WoT needs, especially for searching. For example, information of logical paths and properties of entities and information of the devices that the components simulate or measure are missing. At Intel Berkeley research lab [17], 54 sensors were deployed, and sensor readings were recorded in the form of plain text, which can be used as a dataset for sensor readings.

### B. *WoT Simulations*

Using websites (e.g., [25][26][27]), a WoT environment can be built online by creating channels then attaching them to SThs like Arduinos or other devices equipped with sensors. The devices send information to the attached channels using private keys that are generated by the website. The website receives information from the attached resources to monitor the states of devices or entities that the resources represent. These websites provide RESTful services (GET, PUT, UPDATE, DELETE) [28] for uploading and accessing reading feeds. Moreover, the values (sensor readings) are visualized for users.

The services and design of the aforementioned websites is similar to our proposed testbed environment. However, these websites are limited by available service usage and formats of the responses, which are hardcoded and embedded within website code or at least not exposed to users. The proposed testbed architecture, which is built specially for testing WoT, provides more general services, such as monitoring live information fed from attached SThs, visualizing sensor readings and states of EoIs over time, controlling actuators, triggering action events, and periodic sensor reporting.

### C. *Services Architecture for WoT*

Web services are considered as the main method for accessing WoT devices [15]. Mayer et al. [14] propose a hierarchical infrastructure for building WoT to enhance the performance of the searching service. Nodes receive queries then pass them to the right nodes in the network to answer the queries. The searching scenario starts by getting a list of sensors that can answer a query according to their static properties and predicted values. After that, the identified sensors are queried to check their current values, which are used for ranking the search results. The searching scenario is integrated into the proposed testbed.

Mayer and Guinard [10] and Mayer [29] provide a method for solving the problem of using multiple formats (e.g., microformat and microdata) in the WoT. They propose to add multiple strategies for parsing and producing information in the intended format. However, their work does not result in a dataset. They implemented an algorithm [10], called extensible discovery service, as a Web application that asks users about sensor page URL and retrieves information about devices if and only if the page is written in one of a set of pre-defined formats. Our proposed testbed allows such an algorithm to be tested to measure its performance. The required dataset contains sensor information written in different formats so that the algorithm is tested in parsing and retrieving information about sensors and entities.

To summarize, none of the datasets or testbeds used in the literature fulfills the full requirements for testing and evaluating the Web search process in the WoT as mentioned in Section II. Our proposed testbed environment aims at filling this gap. It is not the main focus of this paper to propose a new WSN testbed. Our main goal is to integrate WoT features above the layer of the IoT for visualizing things and entities, retrofitting on the benefits of existing physical testbeds.

## IV. TESTBED ARCHITECTURE

The proposed testbed architecture transforms the physical control of devices in a surrounding physical environment to an emulated control for those devices keeping the same sense of events and features that existed in the physical environment. These events and features are embedded in datasets that can be later replayed. The

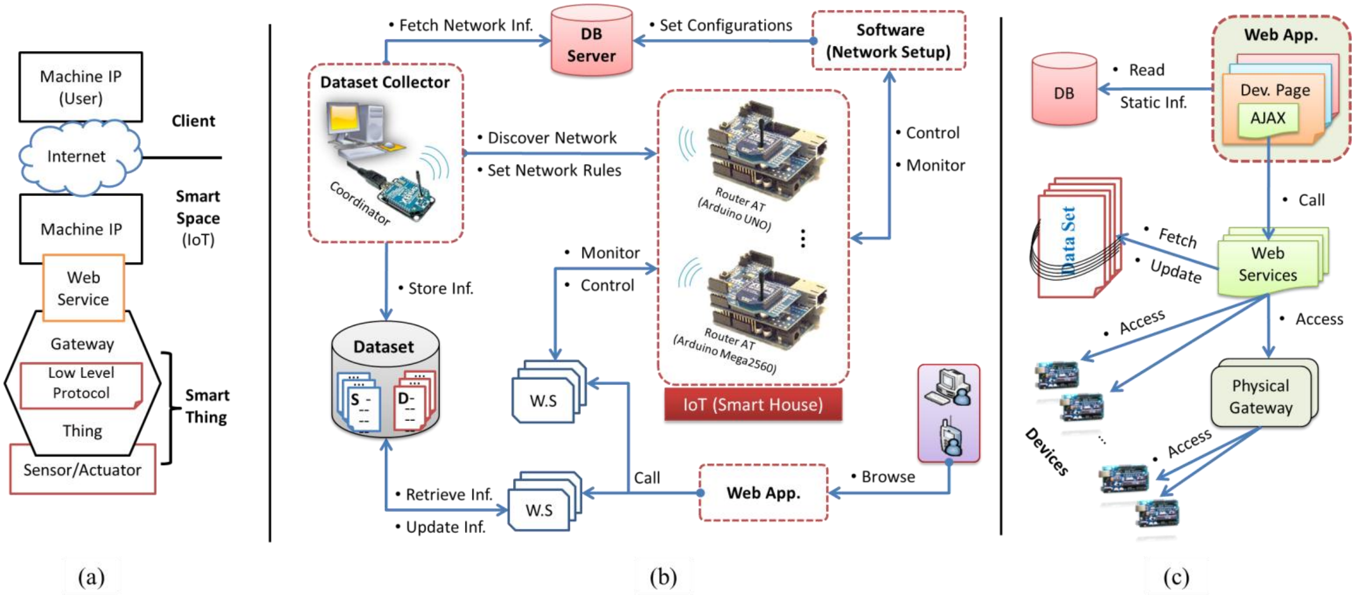


Figure 1. Testbed Architecture: (a) Integrating smart things (STHs) in the IoT - (b) Testbed environment architecture for simulating a physical environment - (c) Web services fetch data from real devices and gateways (online mode), or from dataset files (offline mode).

proposed architecture has two modes of operation: online and offline (Figure 1 (c)). In online mode, datasets are generated, “real” physical information is recorded, and a Web application offers WoT services by accessing the real devices for monitoring and controlling them. In offline mode, the Web application accesses the datasets to replay the events monitoring information.

The testbed architecture, shown in Figure 1 (b), is divided into five parts, as follows**.**

**An IoT infrastructure (e.g., modeling a smart home).** To build the IoT [12], the steps are briefly as follows. First, things are converted to STHs by attaching smart equipment (e.g., sensors and actuators), as shown in Figure 1 (a). Second, the static and dynamic information of STHs is described. STHs representation specifies URLs to invoke STHs services and their parameters and response format [29]. Third, RESTful APIs for accessing the STHs are built. Fourth, communication protocols between STHs and gateways are developed. Fifth, the STHs are connected to the Internet using physical and virtual gateways. STHs integration is done in the form of (1) direct integration, for STHs that support IP address for connection or (2) indirect integration using gateways, for STHs that use low-level protocols [13][30].

**Network setup software**, after building IoT, a program is built for configuring the IoT network. It assigns locations to STHs in the hierarchical structure of the simulated building or environment shown in Figure 2. This allows for using the generated logical path as attributes for the STh.

**Web services for each device** are used for executing WoT services directly and for feeding back users with information about STHs, such as indicated in Figure 1 (a). The web services are hosted on machines that support IP connection, either the STh itself or a physical gateway for accessing STHs that use low-level protocols.

A **Web application** offers WoT services like monitoring and controlling. The application loads information by calling web services, which pull information from devices (online mode) or from WoT dataset files (offline mode), as shown in Figure 1 (c).

The **dataset collector** discovers all available gateways and list of devices connected on each one, sets rules by which

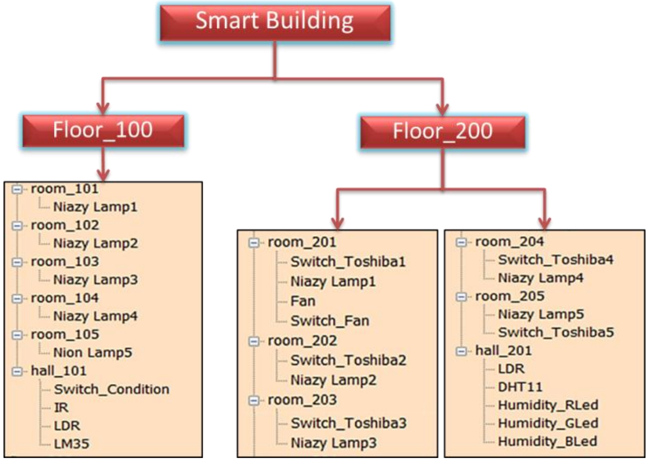
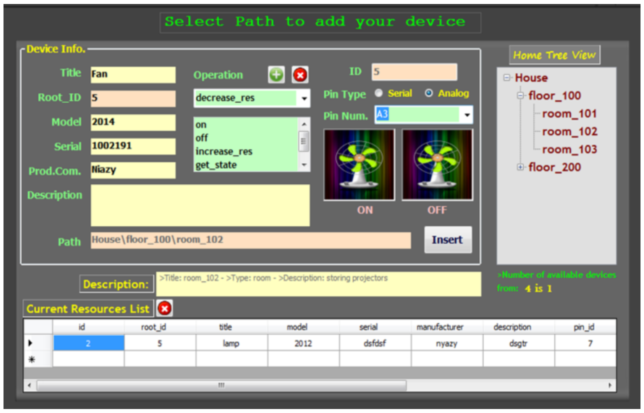


Figure 2. WoT graph for locating devices at specific paths in the hierarchical structure of a building.

ISBN: 978-1-61208-404-6



| | |
|---|---|
| Title | Fan |
| Root_ID | 5 |
| Model | 2014 |
| Serial | 1002191 |
| Prod.Com. | Niazy |
| Description | |
| Operation | decrease_res |
| | on<br>off<br>increase_res<br>get_state |
| ID | 5 |
| Pin Type | ○ Serial ● Analog |
| Pin Num. | A3 |
| Path | House\floor_100\room_102 |

House
floor_100
room_101
room_102
room_103
floor_200

Description: >Title: room_102 - >Type: room - >Description: storing projectors

>Number of available devices from: 4 is 1

| | id | root_id | title | model | serial | manufacturer | description | pin_id |
|---|---|---|---|---|---|---|---|---|
| ▸ | 2 | 5 | lamp | 2012 | dsfdsf | nyazy | dsgtr | 7 |
| * | | | | | | | | |

Figure 3. Locating and configuring a fan device at logical path ‘floor_100\room_102.’

data are collected from them, and sets the format by which the datasets are generated.

## V. Testbed Implementation

According to the testbed architecture presented in the previous section, testbed implementation was done along four axes.

### A. *Building the IoT infrastructure*

This step will be executed the first time around; but, if a dataset that is generated by this testbed exists, then building WoT begins from the next step by attaching the dataset with the web application to work in offline mode.

Building IoT was done in a simple way [31] using widely-available components. The SECE server [23] gets information from the IoT according to events and actions that happen in the environment. It offers the collected information in a friendly user interface. A testbed environment for the WoT is built using these connections.

Building the IoT infrastructure was done in three steps: (1) connecting devices, (2) building the network setup software, and (3) implementing device communication protocols. Whereas it is desirable to build IoT using devices that support direct IP connection rather than devices that support only low-level protocols, the latter devices needed gateways for integrating them into the IoT. The IoT infrastructure was built using Arduinos, on which sensors and actuators were connected. Arduino has two interfaces: a Serial Peripheral Interface (SPI) bus and an Internal Integrated Circuit (I2C), which allows modules, like Ethernet and Secure Digital (SD) cards, to communicate with the microcontroller [32]. The Arduinos connected more than one device using digital and analog pins. In a sense, the Arduinos acted as physical gateways and IP addresses were set for them. They were attached to the network using Ethernet or XBee [33] connections.

**Network setup software** was written in C# for locating, managing and configuring resources for each virtual gateway. A virtual gateway represents a location, such as floor_100 and floor_200. For example, Figure 3 shows the process of adding a new device to the testbed using the software. Logical paths in the building hierarchical structure are very important for accessing devices.

```
void loop(){
   ...
             else if(strcmp(buffer,"GET# ") == 0)
                     send_sr_get(client);
             else if(strcmp(buffer,"POST# ") == 0)
                     send_sr_post(client);
   ...  }
void send_sr_get (EthernetClient client)  {
            // send a standard http response header
             client.println("HTTP/1.1 200 OK");
             client.println("Content-Type: text/html");
             client.print("Value of ");
             client.print(default_Dpin);
             client.print(" is ");
             client.print(digitalRead(default_Dpin));
  ...    }
```

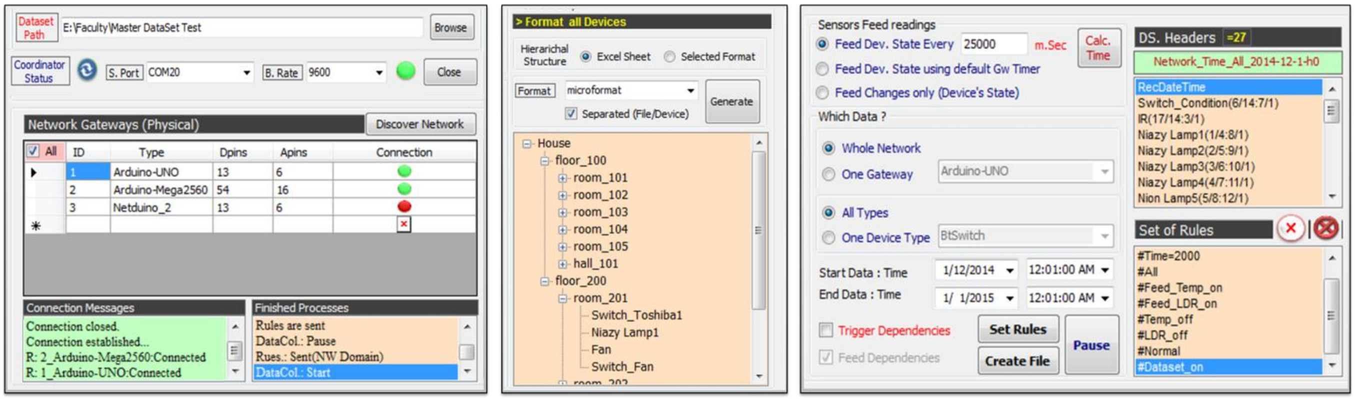
Figure 4. Device network protocol for handling incoming requests of monitoring, controlling, and crawling services (RESTful service).

The **protocols**, written in Arduino Sketches [24], were used to get and set the state of devices that are connected to the Arduinos, whereby get and set requests were sent within the body of the protocol messages. When the special symbol ‘#’ is found within the body of the message, as shown in Figure 4, the spider gets the current device’s states. The crawling case involved only getting information, not controlling or changing device states.

### B. *From IoT to WoT*

Building Web pages in the testbed followed standard features for dealing with dynamic information. The common way for developing dynamic websites depends on AJAX. AJAX is used for live update of some parts in the sensor’s pages. The dynamic parts typically include SThs readings or entity states, which indirectly depend on sensor readings [25][27].

However, pages with dynamic content built using AJAX cannot be crawled by traditional search engine crawlers. Some search engines, such as Google, suggest practical solutions for optimizing the crawling process [34] of dynamic content. Alternative URLs that lead to pages with static information are indexed by default or instead of pages that contain dynamic information. According to Google optimization rules, Web sites in our testbed use AJAX in some parts in device’s web page but for crawling, corresponding Web services are accessed instead to get current STh value or EoI state, in addition to all possible states with corresponding occurrence probabilities. Another technique not implemented in our testbed is to render pages on the fly (i.e., crawlers have browsing processes embedded in their code [35]). Still, it is difficult to crawl pages that need to send some data first before loading their content. Moreover, the time consumed by the crawling process itself becomes high and the crawling process needs to be done



Dataset Path: E:\Faculty\Master DataSet Test [Browse]

Coordinator Status · S. Port: COM20 · B. Rate: 9600 · ● [Close]

Network Gateways (Physical) [Discover Network]

| ☑ All | ID | Type | Dpins | Apins | Connection |
|---|---|---|---|---|---|
| ▶ | 1 | Arduino-UNO | 13 | 6 | 🟢 |
|  | 2 | Arduino-Mega2560 | 54 | 16 | 🟢 |
|  | 3 | Netduino_2 | 13 | 6 | 🔴 |
| * |  |  |  |  | ☒ |

Connection Messages:
Connection closed.
Connection established...
R: 2_Arduino-Mega2560:Connected
R: 1_Arduino-UNO:Connected

Finished Processes:
Rules are sent
DataCol.: Pause
Rues.: Sent(NW Domain)
DataCol.: Start

(a)

> Format all Devices

Hierarichal Structure: ● Excel Sheet ○ Selected Format

Format: microformat · ☑ Separated (File/Device) [Generate]

```
⊟ House
   ⊟ floor_100
      ⊞ room_101
      ⊞ room_102
      ⊞ room_103
      ⊞ room_104
      ⊞ room_105
      ⊞ hall_101
   ⊟ floor_200
      ⊟ room_201
         Switch_Toshiba1
         Niazy Lamp1
         Fan
         Switch_Fan
      ⊞ room_202
```

(b)

Sensors Feed readings
● Feed Dev. State Every 25000 m.Sec [Calc. Time]
○ Feed Dev. State using default Gw Timer
○ Feed Changes only (Device's State)

Which Data ?
● Whole Network
○ One Gateway: Arduino-UNO
● All Types
○ One Device Type: BtSwitch

Start Data : Time 1/12/2014 12:01:00 AM
End Data : Time 1/ 1/2015 12:01:00 AM

☐ Trigger Dependencies
☑ Feed Dependencies
[Set Rules] [Create File] [Pause]

DS. Headers =27
Network_Time_All_2014-12-1-h0
RecDateTime
Switch_Condition(6/14:7/1)
IR(17/14:3/1)
Niazy Lamp1(1/4:8/1)
Niazy Lamp2(2/5:9/1)
Niazy Lamp3(3/6:10/1)
Niazy Lamp4(4/7:11/1)
Nion Lamp5(5/8:12/1)

Set of Rules
#Time=2000
#All
#Feed_Temp_on
#Feed_LDR_on
#Temp_off
#LDR_off
#Normal
#Dataset_on

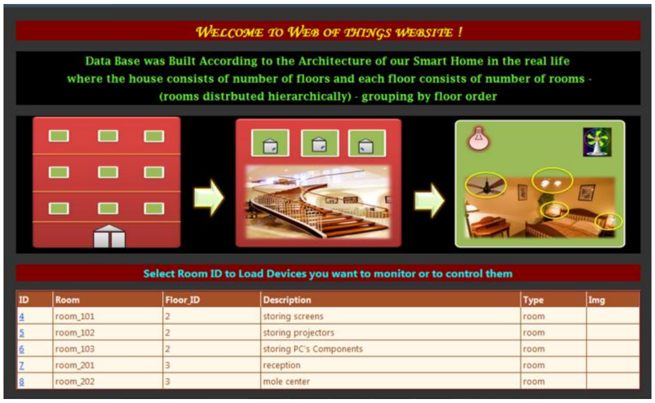
(c)

Figure 5. Dataset collector software: (a) Dataset collector program discovers gateways in WoT loading a list of devices connected on them. (b) Static information about the IoT testbed are generated in different formats. (c) Rules are defined to control the way gateways send dynamic information (readings).

frequently; information in WoT may be updated in less than a minute.

Using Ethernet, RESTful APIs can access Arduino components. Devices are programmed as *clients* to push sensor data to some services and as *services* to enable remote control of devices over the Web. Because it is desirable to have a Web page for each device and because Arduino acts as a gateway for managing at least one component, a website is built and can be hosted on an SD card connected to the Arduino. The website is accessed using an IP address, assigned to the Arduino. Another alternative is to host the website on a different server for adding more capabilities like storage capacity. In the latter case, Arduinos are accessed using RESTful APIs. The selection of either alternative is determined by the amount of information that needs to be stored and accessed over time.

Two steps were done to add WoT layer to the testbed. First, a **Web application** was written in Asp.Net (Figure 6). The main services of the Web application are monitoring sensors, controlling actuators, triggering action events, and periodic sensor reporting [12][23][30]. The WoT application was built according to the building hierarchical structure configured by the network setup software. The homepage shows general information and allows users to perform general tasks, such as monitoring room status. The user selects a logical path to browse, then, for each room, a list of devices and their states appear. The user selects a device to access. The device page loads the RESTful services dynamically (using WSDL [36]) according to the Arduino IP and selected device ID. Second, a set of **Web services** were written in C#. The Web application loads the available RESTful services dynamically for each device. A special tag *'GET#'* is added as an additional service that is executed by default for the device webpage. The crawling process returns the current sensor value or the state of the device and all possible states with their probabilities.



WELCOME TO WEB OF THINGS WEBSITE !

Data Base was Built According to the Architecture of our Smart Home in the real life where the house consists of number of floors and each floor consists of number of rooms - (rooms distrbuted hierarchically) - grouping by floor order

Select Room ID to Load Devices you want to monitor or to control them

| ID | Room | Floor_ID | Description | Type | Img |
|---|---|---|---|---|---|
| 4 | room_101 | 2 | storing screens | room |  |
| 5 | room_102 | 2 | storing projectors | room |  |
| 6 | room_103 | 2 | storing PC's Components | room |  |
| 7 | room_201 | 3 | reception | room |  |
| 8 | room_202 | 3 | mole center | room |  |

Figure 6. Web pages of virtual gateways get their information from a database server. Sensor Web pages get their information either from direct access to devices or from the offline dataset.

### C. Dataset Collector (DsC)

In Figure 5 (a), using Zigbee connection, the WoT coordinator (gateway that acts as a base station) discovers all available gateways, getting a list of connected devices on each gateway. The dataset collector (DsC) program generates files written in different formats for the static information of the IoT testbed including the building hierarchy and the devices located in the hierarchy (Figure 5 (b)). The dynamic information is collected using a set of rules, as in Figure 5 (c). The rules instruct the gateways to send back specific information about a specific list of devices according to a specific action or event done by other devices. The gateways feed the DsC with device readings according to these rules. If the rule '*ChangesOnly*' is selected, the DsC stores only changes on device state. If the rule '*TimeSlot*' was selected, the DsC stores periodical feeds of device state. One of the most important rules is that if a certain device type is selected for analysis of device readings and making decisions according to the analysis results, rules can be set to collect data from all devices of that type across all the gateways in WoT.

Copyright (c) IARIA, 2015.     ISBN:  978-1-61208-404-6

```
<div class ="hproduct ">
        <span class =" fn">22_Fan</span>
        <span class =" identifier">,
             <span class =" type ">Fan</span> Sfan123
             <span class =" value ">0</span>
        </span>
        <span class =" category ">
             <a href =http://www.XXX.com   rel =" tag"> Fan </a>
        </span>
        <span class =" brand ">Brand Name</span>
        <span class =" description "> characterized by …
        </span>
        <span class =" Photo ">Fan</span>
             <a href =http://www.XXX.com/?s=wsn
                  class =" URL">
                          More information about this device.
              </a>
</div>
```

Figure 7. Static information about a fan written in microformat.

### D. Generated Dataset Files

A simple dataset was generated by the testbed according to the rules: (1) '*every_2500 msc*' for updating dataset every 2500 millisecond (i.e., DsC pulls information from the network), it could be replaced by rule '*ChangesOnly*' for storing changes on devices' states only (i.e., devices push information to DsC), (2) '*All_Network*' for pulling information from all discovered gateways in the network, (3) '*All_types*' means all devices on selected gateways, (4) '*2014-12-1-h0_to_2015-1-1-h0*' for storing dataset from '1/12/2014' to '1/1/2015', and (5) '*TD*' for triggering all dependences related to selected devices. The dataset generated according to limited time slot (date and time) by DsC, as shown in Figure 5 (b), contains static information about IoT infrastructure and dynamic information about sensing and actuating activities.

The static information of each device, such as logical path and device type, is stored in a file named using the device ID, the EoI ID, and device name (e.g., *22_9_Fan*). Static information about a fan written in microformat is shown in Figure 7.

The dynamic information, such as sensor readings, is stored in a file named using the collection-rule title and the date and time of collection (e.g., *Network_Time_All_2014-12-1-h0*). This file contains readings collected from all devices in the WoT testbed. A subset of data stored in that file would look like Table 1, where monitoring is set to rule '*time only'*.

As mentioned before, sensor definition contains information about sensor so that it can be accessed easily through built web application that simulate the WoT. In sensor definition: '*Xlamp2(2/5:9/1)*', X is sensor's brand name, lamp2 is sensor title, (2/5) is the virual location sensor id and hosting room id, and (9/1) is the physical location where 9 is pin number and 1 is gateway id. Column '*Time*' is the response time. Arduinos support 5 Voltages as maximum; they convert voltage range (0:5V) to be (0:255) using built in analog to digital converter. Values recored for each sensor definition, are current voltages consumed by that sensor (0 : 255).

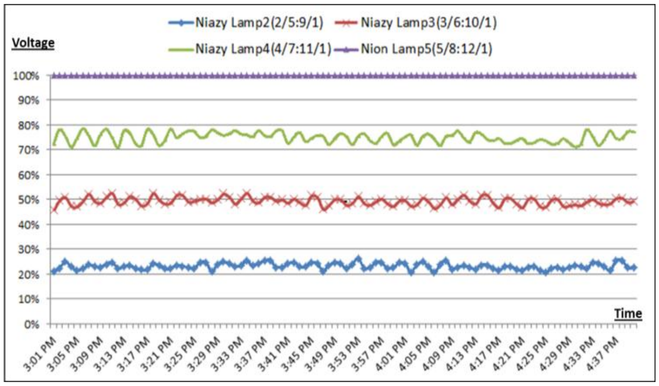


Figure 8. A graph generated out of a dynamic data file collected from devices of the same type (lamp in this graph). Voltages consumed by devices are represented in percentage at Y axis.

TABLE 1. SAMPLE READINGS GENERATED FROM ALL SENSORS OF TYPE 'LAMP' IN THE ENTIRE NETWORK AS A TIME SERIES. CONSUMED VOLTAGES ARE MAPPED FROM (0:5) TO (0:255).

| **Time** | **XLamp2 (2/5:9/1)** | **XLamp3 (3/6:10/1)** | **XLamp4 (4/7:11/1)** | **XLamp5 (8/8:12/1)** |
|---|---|---|---|---|
| 03:01 PM | 66 | 77 | 83 | 71 |
| 03:02 PM | 66 | 80 | 83 | 68 |
| 03:04 PM | 66 | 68 | 65 | 69 |
| 03:06 PM | 67 | 71 | 69 | 67 |
| 03:08 PM | 68 | 80 | 85 | 70 |
| 03:10 PM | 69 | 81 | 86 | 70 |
| … | … | … | … | … |
| 10:12 PM | 65 | 80 | 85 | 70 |

## VI. CASE STUDY

In this section, a case study of the proposed WoT testbed is described.

### A. Using WoT Dataset for information analysis

Using the generated dataset, researchers can analyze sensor data collected using multiple controlled scenarios. A lot of experiment scenarios can be achieved on the testbed, such as comparing the state of devices on certain gateways (e.g., gateways of a room), comparing state of devices on all gateways (e.g., all gateways of a building), getting live time of each device and high level power consumption in daily live to provide suggestions related to energy efficiency achievement (i.e., Energy awareness through interactive user feedback). Figure 8 shows a comparison between devices of type '*Lamp*' for analysis of power consumption. Y-axis represents consumed voltages percent and X-axis represents time. Estimation on timing accuracy of the data hasn't been measured yet, but after enlarging WoT scale, DsC can estimate timing accuracy by calculating request and receive time for each device. In general, such a dataset, especially composed of dynamic information, will be helful for computing Fuzzy-based sensor similarity [15], and for running prediction algorithms on real information that are used in searching about SThs and EoIs in the WoT.

```
Select   [Device_Header]
From     [Sheet_Title]
Where    [RecDateTime] = (Select      min ([RecDateTime])
                          From        [Sheet_Title]
                          Where       [RecDateTime]
                          Between     @Date_1 and @Date_2)
```

Figure 9. Accessing dataset files using web services (offline mode): Selecting column ‘Device_Header’ from sheet ‘Sheet Title’ where its time = current system time (hours and minutes) using OleDbCommand.

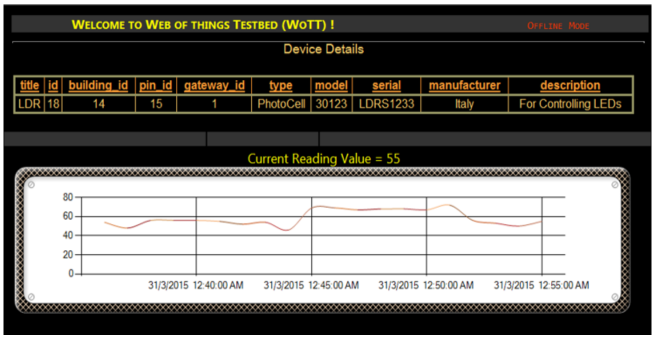
Providing information about SThs and EoIs in multiple formats with additional attributes like logical paths expands experimental work in this area.

### B. *Browsing WoT*

Building simple and physical WoT (offline or online) will be helpful and more accurate than using simulators. Figure 1 (c) shows a scenario of calling RESTful web services for pulling information about buildings and their devices from the generated dataset (offline). Sensor pages call Web services that fetch information from a dataset file *‘Network_Time_All_2014-12-14-h17.xlsx’* using command of type *OleDbCommand*. Web services called in the testbed (Figure 1 (c)) execute the command string shown in Figure 9. *‘Device_Header’* and *‘Sheet_Title’* were sent by calling pages to the Web service *monitoring*. The special character @ before variables *‘Date_1’* and *‘Date_2’* means that they are initiated within the Web service.

### C. *Reusing Testbed for Different IoT*

The proposed testbed architecture allows the implementation of different purposes in the WoT. If someone has to operate the testbed for a certain environment (for example, energy saving of smart home, detect something unacceptable happening at a shopping mall, etc.), and because the proposed testbed operates in two modes (online and offline), then reusing this testbed is restricted with operation mode; For online mode, new IoT infrastructure, which is built by attaching resources support information about measuring physical phenomena and actuating EoIs, is replaced by the IoT part shown in Figure 1 (b) and registered by **Network setup software**. New IoT should speak the same language as the DsC (gateways make it easy for supporting heterogeneity in devices); But for offline mode, such as shown in Figure 10, because the dataset represents the IoT itself where it hosts information about SThs, EoIs, and sensing and actuating processes, then IoT part will be replaced by that dataset to be accessed by web services as indicated in Figure 1 (c), so Offline mode could be used for retesting previously built IoT.



WELCOME TO WEB OF THINGS TESTBED (WOTT) !

OFFLINE MODE

Device Details

| title | id | building_id | pin_id | gateway_id | type | model | serial | manufacturer | description |
|---|---|---|---|---|---|---|---|---|---|
| LDR | 18 | 14 | 15 | 1 | PhotoCell | 30123 | LDRS1233 | Italy | For Controlling LEDs |

Current Reading Value = 55

Figure 10. Testbed is running in offline mode (attaching IoT dataset).

## VII. CONCLUSION AND FUTURE WORK

WoT has become one of the most trendy research directions due to facilities and services provided in many domains. Sensors can provide great benefits if their readings are presented in a meaningful and friendly way to users and machines. Searching for SThs and EoIs is one of the most important services in the WoT. But, most of the work in this area focuses on searching in a single environment. In other words, the WoT is built using single formatting and network infrastructure. In this work, a WoT testbed is proposed to be built in a simple way with readily-available physical components. The proposed testbed allows capturing - and maintaining for offline usage - the sense of events and actions in the environment under test. The testbed allows for building a WoT environment according to a hierarchical architecture, providing description for components in a way that gives search engine spiders the ability to crawl them in addition to the ability given to users to perform live monitoring of their environment. The dataset generated from the testbed is expected to help research on the crawling, indexing, and searching processes in WoT in general.

The problem of searching about SThs depends on the standardization of formats used for representing SThs (properties and services they offer). So, providing semantic discovery services based on application of multiple discovery strategies [10] and enriching SThs metadata may enhance results of searching and lookup services in the WoT. Creating standardized RESTful service description embedded in HTML representation using microdata is feasible and desirable [29]. Still a few important questions remain here: what is the timing accuracy of the data, what information needs to be indexed, and how to index WoT data streams.

## REFERENCES

[1] M. Blockstrand, T. Holm, L.-Ö. Kling, R. Skog, and B. Wallin, "Operator opportunities in the internet of things – getting closer to the vision of more than 50 billion connected devices," [Online] Feb. 2011, http://www.ericsson.com/news/110211_edcp_244188811_c, ,(accessed: 10 Feb. 2015)

[2] L. Coetzee and J. Eksteen, "The Internet of Things – Promise for the Future ? An Introduction," in IST-Africa Conference, Gaborone, May 2011, pp. 1-9.

[3] Ch. Lerche, "WS4D-uDPWS - The Devices Profile for Web Services (DPWS) for highly resource-constrained devices," WS4D Initiative, [Online] Aug. 2010, http://code.google.com/p/udpws/wiki/IntroductionGeneral, (Accessed: 10 April 2015).

[4] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: A Wireless Sensor Network Testbed," in Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on, Boise, ID, USA, 2005, pp. 483-488.

[5] H. Sundani, H. Li, V. K. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless Sensor Network Simulators - A Survey and Comparisons," International Journal Of Computer Networks (IJCN), vol. 2, no. 6, pp. 249-265, Feb. 2011.

[6] I. Muhammad, A. Md Said, and H. Hasbulla, "A Survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks," in nformation Technology (ITSim), 2010 International Symposium in, vol. 2, Kuala Lumpur, June 2010, pp. 897-902.

[7] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo, "A Survey on Facilities for Experimental Internet of Things Research.," IEEE Communications Magazine, Institute of Electrical and Electronics Engineers (IEEE), no. <10.1109/MCOM.2011.6069710>. <inria-00630092>, pp. 58-67, 2011, 49 (11).

[8] B. Ostermaier, B. M. Elahi, K. Römer, M. Fahrmair, and W. Kellerer, "A Real-Time Search Engine for the Web of Things," in The 2nd IEEE International Conference on the Internet of Things (IoT), Tokyo,Japan, Nov. 2010., pp. 1-8.

[9] Shodan, "The search engine for the Internet of Things, " [Online] 2015, https://www.shodan.io/, (Accessed: 10 April 2015).

[10] S. Mayer and D. Guinard, "An Extensible Discovery Service for Smart Things," in Proceedings of the 2nd International Workshop on the Web of Things (WoT 2011), ACM, San Francisco, CA, USA, June 2011, pp. 7-12.

[11] S. Haller, "The Things in the Internet of Things," Poster at the (IoT 2010). Tokyo, Japan, vol. 5, no. 26, p. 4, Nov. 2010.

[12] D. Guinard, "A Web of Things Application Architecture - Integrating the Real-World into the Web," PhD Thesis, Computer Science, Eidgenössische Technische Hochschule ETH Zürich, Zürich, 2011.

[13] D. Guinard, V. Trifa, S. Karnouskos, and D. Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services," Services Computing, IEEE Transactions on, vol. 3, no. 3, pp. 223-235, Sep. 2010.

[14] S. Mayer, D. Guinard, and V. Trifa, "Searching in a Web-based Infrastructure for Smart Things," in Proceedings of the 3rd International Conference on .,the Internet of Things (IoT 2012),IEEE, Wuxi, China, October 2012, pp. 119-126.

[15] C. Truong, K. Romer, and K. Chen, "Sensor Similarity Search in the Web of Things," in In World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium, San Francisco, CA, June 2012, pp. 1-6.

[16] J. Miloš , N. Zogović, and G. Dimić, "Evaluation of Wireless Sensor Network Simulators," in the 17th Telecommunications Forum (TELFOR 2009), Belgrade, Serbia, 2009, pp. 1303-1306.

[17] P. Bodik, C. Guestrin, W. Hong, S. Madden, M. Paskin, and R. Thibaux. "Intel Lab Data," [Online] Apr. 2004, http://www.select.cs.cmu.edu/data/labapp3/index.html, (Accessed: 10 April 2015).

[18] D. Gay, P. Levis, D. Culler, E. Brewer, M. Welsh, and R. von Behren, "The nesC language: A holistic approach to networked embedded systems," in PLDI '03 Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation, New York, NY, USA, May 2003, pp. 1-11.

[19] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, and A. Woo, "TinyOS: An Operating System for Sensor Networks," in Ambient Intelligence, W. Weber, J. M. Rabaey, and E. Aarts, Eds. Springer Berlin Heidelberg, 2005, ch. 2, pp. 115-148.

[20] Hosted by Thingsquare, "Contiki: The Open Source OS for the Internet of Things, " [Online] 2012, http://www.contiki-os.org/, (Accessed: 10 April 2015).

[21] M. Nati, A. Gluhak, H. Abangar, and W. Headley, "SmartCampus: A user-centric testbed for Internet of Things experimentation," in Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on, Atlantic City, NJ, June 2013, pp. 1-6.

[22] G. Mujica, V. Rosello, J. Portilla, and T. Riesgo, "Hardware-Software Integration Platform for a WSN Testbed Based on Cookies Nodes," in IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society , Montreal, QC, October. 2012, pp. 6013-6018.

[23] H. Nam, J. Janak, and H. Schulzrinne, "Connecting the Physical World with Arduino in SECE," Computer Science Technical Reports, Department of Computer Science, Columbia University, New York, Technical Reporting CUCS-013-13, 2013.

[24] Arduino, "Arduino," [Online] 2015, http://www.arduino.cc/, (Accessed: 10 April 2015).

[25] LogMeIn, Inc., "Xively," [Online] 2014, http://www.Xively.com, (Accessed: 10 April 2015).

[26] XMPro, "Intelligent Business Operations Suite For The Digital Enterprise," [Online] 2015, http://xmpro.com/xmpro-iot/, (Accessed: 10 April 2015).

[27] IoBridge, "ThingSpeak- The open data platform for the Internet of Things," [Online] 2015. http://www.thingspeak.com, (Accessed: 10 April 2015).

[28] M. Elkstein, "Learn REST: A Tutorial," [Online] 2008. http://rest.elkstein.org/2008/02/what-is-rest.html, (Accessed: 10 April 2015).

[29] S. Mayer, "Service Integration - A Web of Things Perspective," in W3C Workshop on Data and Services Integration, Citeseer, Bedford, MA, USA, October 2011, pp. 1-5.

[30] D. Guinard and V. Trifa, "Towards the Web of Things: Web mashups for embedded devices," in in Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain, 2009, p. 15.

[31] C. Pfister, "The Internet of Things," in Getting Started with the Internet of Things: Connecting Sensors and Microcontrollers to the Cloud, B. Jepson, Ed. United States of America.: O'Reilly Media, Inc., 2011, ch. 4, pp. 29-41.

[32] A. McEwen and H. Cassimally, "Designing the Internet of Things," 1st ed., C. Hutchinson, Ed. John Wiley & Sons, ISBN: 1118430638;9781118430637, November 2013, https://books.google.com.eg/books?id=oflQAQAAQBAJ, (Accessed: 10 April 2015).

[33] Digi International Inc., "Official XBee website- Connect Devices to the Cloud," [Online] 2015. http://www.digi.com/xbee, (Accessed: 10 April 2015).

[34] Google, "Search Engine Optimization (SEO) - Starter Guide," Jan. 2010.

[35] P. Suganthan G C, "AJAX Crawler," in Data Science & Engineering (ICDSE), 2012 International Conference on. IEEE, Cochin, Kerala, July 2012, pp. 27-30.

[36] wikipedia, "Web Services Description Language," [Online] Apr. 2015, http://en.wikipedia.org/wiki/Web_Services_Description_Language , (Accessed: 10 April 2015).

# MCAST: Mobility-aware Channel-Availability based Channel Selection Technique

Md. Arafatur Rahman and Mohammad Moshee Uddin

Faculty of Computer Systems and Software Engineering
University Malaysia Pahang
Gambang, Malaysia
Email: arafatur@ump.edu.my, mdmoshi@yahoo.com

Roberto Savoia

Istituto di Cibernetica ”E. Caianiello”
Naples, Italy
Email: roberto.savoia@ino.it

***Abstract*—A key issue in cognitive radio networks is the design of a channel selection technique that guarantees to utilize the highest available channel in presence of the dynamic activity of primary users. Usually, the channel selection techniques that operate in this kind of network are based on the channel-availability probability. In the static primary user's scenario, this probability can be *a priori* known or simply estimated from the channel occupancy history. However, in the mobile primary user's scenario, this probability dynamically varies in time due to the changes of the primary user's position. In order to exploit the dynamic variation of the channel availability, in this paper we design a novel Mobility-aware Channel-Availability based channel Selection Technique (MCAST) that ensures the selection of the channel with the highest channel availability probability in a given temporal interval. The simulation results highlight the benefits of the proposed technique in presence of primary user's mobility. Moreover, we evaluate the effectiveness of MCAST in a scenario of practical interest by adopting this technique in a recently proposed routing metric designed for this network.**

***Index Terms*—Cognitive Radio, PU Mobility, Channel Availability Probability.**

## I. Introduction

In Cognitive Radio Networks (CRNs), the channel selection techniques are usually based on the knowledge of the Channel-Availability Probability (CAP), i.e., the probability that the channel is available for the unlicensed users, referred to as Cognitive Users (CUs), without causing interference against the licensed users, referred to as Primary Users (PUs). In fact, this knowledge enables the CU to select the channel with the highest availability. Usually, the CAP coincides with the probability that at a certain time the PU is inactive, and can be *a priori* known or simply estimated from the channel occupancy history [1]. However, this assumption is valid when the PU is static.

On the other hand, in the mobile PU scenario, the CAP dynamically varies in time due to the changes of the PU position. For instance, if at a certain time the CU is outside the *protection range* (i.e., it is defined as the maximum distance between the PU and the CU at which the CU transmission does not interfere the PU communication on an arbitrary channel. It is determined by the PU transmission range and by the CU interference range [2]) of an arbitrary PU, then the CAP is independent from whether the PU is inactive or not. Due to the PU mobility, after a certain interval of time, the CU might be inside the protection range of the PU, then the CAP depends on the probability that the PU is inactive. Since the best performance is guaranteed by the channel with the highest CAP assumed at a given time, a fundamental key issue in CRNs is the *design of a channel selection technique that ensures the selection of the best channel by exploiting the dynamic variation of the channel availability caused by the PU mobility*.

Basically, most of the works in literature consider the static PU scenario where the CAP does not vary in time. In [3], Jha et al. propose an opportunistic multi-channel Medium Access Control (MAC) with QoS provisioning for distributed CRNs, where CUs use the previous channel scanning results to select those channels with the highest CAP. In [4], Xue et al. propose an opportunistic periodic MAC protocol where the CUs cooperate each wit other to share the channel-availability information. In [1], Chowdhury et al. propose a routing metric that aims to minimize the interference caused by the CUs against the static PUs. In [5], Caleffi et al. propose an optimal routing metric for CRNs where the channel is selected based on channel occupancy history. Finally, there are some other channel selection strategies that have been proposed in literature by using the assumption of static PU activity [6][7][8].

However, the design of a channel selection technique that accounts for the CAP in presence of PU mobility has not yet been addressed in literature. Cacciapuoti et al. [2] addressed the concept of CAP in mobile scenario. For this reason, we design a novel Mobility-aware Channel-Availability based channel Selection Technique (MCAST) that ensures the selection of the channel with the highest CAP in a given temporal period.

Specifically, the contribution of this work can be summarized as follows. First, we derive the channel-availability estimation method in presence of PU mobility. Then, we prove that the proposed channel selection technique takes advantage of the dynamic variation of channel-availability caused by the PU mobility and, consequently, outperforms the traditional method which is only based on the PU temporal activity. The simulation results highlight the benefits of the

ISBN: 978-1-61208-404-6

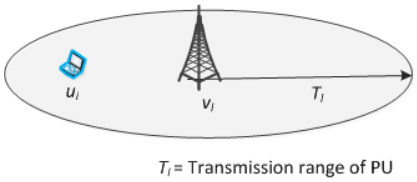


Fig. 1: CAP in presence of static PU.

proposed technique. Moreover, we evaluate the effectiveness of MCAST in a scenario of practical interest by adopting this technique in a recently proposed routing metric, referred to as Optimal Primary-aware routE quAlity (OPERA) [5], designed for CRNs.

The rest of the paper is organized as follows. In Section II, we introduce the problem statement. In Section III, we discuss about the network model, while in Section IV, we describe the channel-availability estimation process. We describe the proposed MCAST in Section V, while the performances are evaluated through simulations in Section VI. Finally, in Section VII, we conclude the paper.

## II. Problem Statement

In this section, we describe how the CAP in static PU scenario differs from the Mobile PU scenario and then, we present our proposal to overcome the adverse effects of PU mobility.

### A. Static PU Scenario

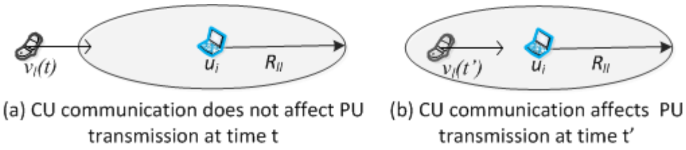
In the static PU scenario, the geographic location of each PU is fixed, as shown in Figure. 1. In this case, the channel selection strategy, referred to as *static method*, considers PU inactive probability, denoted as $P^m_{off}$, for selecting the channel with the highest CAP [3][4][5]. This probability can be *a priori* known or simply estimated based on the channel occupancy history [1]:

$$P^m_{off} = \frac{\alpha^m}{\alpha^m + \beta^m} \tag{1}$$

where $\frac{1}{\alpha^m}$ and $\frac{1}{\beta^m}$ are the average *on* and *off* times for the *m-th* channel, respectively. The *on* time refers to the period where the *m-th* channel is occupied by the PU, while the *off* time indicates the channel is free for CU transmission.

### B. Mobile PU Scenario

On the other hand, in the mobile PU scenario, the geographic location of each PU is not fixed and the channel availability dynamically varies in time. In fact, in Figure. 2 (a), the transmission of the *i-th* CU, denoted as $u_i$, does not affect the PU receiver at time $t$, since $u_i$ is outside the *protection range* of the *l-th* PU transmitter, denoted as $v_l$. However, $v_l$ is moving toward $u_i$ at time $t$, then after a certain interval of time, in Figure. 2 (b), the transmission of $u_i$ might affect the PU receiver at time instant $t'$, since $u_i$ is inside the protection range of $v_l$. Therefore, the CAP varies in the interval $[t, t']$.



(a) CU communication does not affect PU transmission at time t

(b) CU communication affects PU transmission at time t'

Fig. 2: CAP in presence of mobile PU.

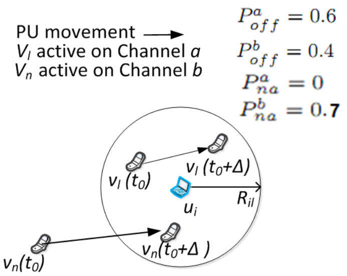


Fig. 3: Static method fails in selecting the channel with highest CAP between *a* and *b*. It selects channel *a*, although channel *b* is the highest one.

In this scenario, if the CU selects the channel according to the *static method*, it will not achieve the channel with the highest CAP. We discuss this issue with an example. As shown in Figure. 3, there are two PUs, denoted as $v_l$ and $v_n$, which are communicating on channel *a* and *b*, respectively. Due to the PU mobility, in a certain interval of time $[t_0, t_0 + \Delta]$, the CAP depends on two factors: *i)* The PU inactive probabilities, i.e., $P^m_{off}$; *ii)* The probability that the CU transmission does not affect the PU while it is active, i.e., $P^m_{na}$. The *static method* selects the best channel considering only the first factor. Since $P^a_{off}$ is greater than $P^b_{off}$, the best channel with the highest CAP according to the *static* method is channel *a*, since $P^a_{off} > P^b_{off}$. However, in presence of PU mobility, the selection of channel *a* does not assume the best choice in terms of channel availability. According to the procedure which considers both factors, referred to as *mobility-aware method*, the channel with the highest CAP is channel *b*, since in this method the $CAP^a = P^a_{off} + (1 - P^a_{off})P^a_{na} = 0.6 + 0.4 \times 0 = 0.6$ is less than $CAP^b = P^b_{off} + (1 - P^b_{off})P^b_{na} = 0.4 + 0.6 \times 0.7 = 0.82$. As a result, at a certain time $t_0$, the *mobility-aware method* achieves the best channel in presence of PU mobility selecting it with the highest CAP for the next interval of time $[t_0, t_0 + \Delta]$.

From the above example, it is evident the need for designing a proper channel selection technique that ensures the selection of the best channel by exploiting the dynamic variation of the CAP caused by the PU mobility.

### C. Proposed Method

The proposed channel selection technique is based on the channel-availability estimation method in order to estimate the channel-availability for a given interval of time. The method is described as follows.

- Firstly, it estimates the distance between PU and CU at time instant $t$ where $t$ belongs to the next temporal interval (See Section IV-A);

- Based on the estimated distance, it estimates the CAP for each channel at time $t$ ;
- Then, it estimates the average CAP for each channel in the next temporal interval;
- Finally, MCAST selects the channel based on the highest average CAP for the next temporal interval.

The proposed channel selection technique is designed for mobile PU scenario with the objective to overcome adverse effects of PU mobility.

## III. Network Model

In this section, we describe the PU and CU network model.

### A. PU Network Model

The PUs move according to the Random WayPoint Mobility (RWPM) model [9] inside a square network region $\mathcal{A}$. Each PU randomly chooses a destination point in $\mathcal{A}$ according to a uniform distribution, and it moves towards this destination with a velocity modeled as a random variable uniformly distributed in $[v_{min}, v_{max}]$ $m/s$ and, statistically independent of the destination point. During each PU movement period, it is assumed that the PU does not change its direction and velocity. The PU traffic on the *m*-th channel is modeled as a two-state birth-death process [10]. Moreover, we consider two different PU spectrum occupancy models [11]. In the first model called Single PU for Channel (SPC), the PUs roam within the network region using different channels. In the second model called Multiple PUs for Channel (MPC), different mobile PUs can use the same channel.

### B. CU Network Model

The CUs are assumed static (it is straightforward to prove that the derived expressions hold also if we assume mobile CR users and static PUs), where $u_i(t)$ denotes the position of the *i-th* CU that is constant in time. Each CU obtains its location information once during the network initialization, (the CU can obtain its location either directly through dedicated positioning systems such as Global Positioning System (GPS) or indirectly through location estimation algorithms) whereas it can update the PU position every $\tau$ seconds, referred to as *PU position updating interval*. It is reasonable to assume that the CU cannot access the PU location in each time instant t, since the PU location is time-variant and it is obtained through either location estimation algorithms [12][13][14] or dedicate databases.

## IV. Channel-Availability Estimation

Since the CU does not know the effective PU position during $\tau$, a distance estimation method should be derived to estimate the channel availability during this temporal interval, with the aim to select the channel with the highest CAP. Thus, in this section we single-out the distance estimation procedure (Section. IV-A) and the estimated CAP expression in both Single PU for Channel (Section. IV-B) and Multiple PUs for Channel (Section. IV-C) scenarios. Finally, we discuss the trade-off that exists between the PU position updating interval $\tau$ and the distance estimation error (Section. IV-D). The CAP estimation process of [2] is adopted in this paper. For the details about CAP estimation, we refer the reader to [2].

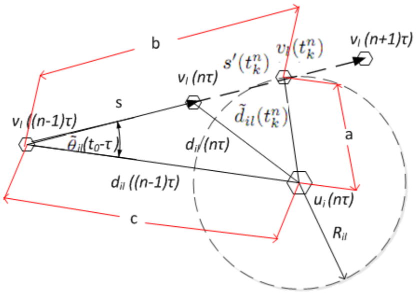


Fig. 4: Distance Estimation Procedure.

### A. Distance Estimation Procedure

The distance estimation procedure is depicted in Figure. 4 where the *i-th* CU and the *l-th* PU are denoted with $u_i$ and $v_l$, respectively, and $R_{il}$ denotes the *protection range*. The PU $v_l$ is mobile and its position at a generic time instant $t$ is denoted, for simplicity of notation, as $v_l(t)$, and the distance between $u_i$ and $v_l$ is denoted as $d_{il}(t)$. At the time instant $t_0$, the CU can calculate several parameters related to the previous interval $[t_0-\tau,\, t_0]$, such as $d_{il}(t_0-\tau)$ and $d_{il}(t_0)$, the traveled distance of $v_l$ during the interval $[t_0-\tau,\, t_0]$ denoted as $s$, the estimated movement direction of $v_l$ toward $u_i$ at time $(t_0-\tau)$ denoted as $\tilde{\theta}_{i,l}(t_0-\tau)$. Based on these information, along with the estimated traveled distance of $v_l$ during the interval $[t_0,\, t]$ denoted as $s'(t)$, $u_i$ can estimate the distance $\tilde{d}_{il}(t)$ when $t$ belongs to the next temporal period $[t_0,\, t_0+\tau]$.

### B. CAP estimation in Single PU for Channel (SPC) scenario

In this subsection, we derive (Theorem 1) the estimation $\tilde{p}^m_{il}(t)$ of the CAP when $t \in [t_0, (t_0+\tau)]$ in SPC scenario under the following assumptions: i) The CU knows the PU position at the actual time instant $t_0$ and at the previous time instant $t_0-\tau$; ii) The PU does not change its direction and velocity during the interval $[t_0-\tau, t_0+\tau]$ (this assumption is reasonable beacuse of the PU mobility model). Since the proof of Theorem 1 requires an intermediate result, we first present it in Lemma 1.

**Lemma 1.** *The estimated distance $\tilde{d}_{il}(t)$ between the i-th CU and the l-th PU at time instant $t \in [t_0, (t_0+\tau)]$ is given by:*

$$\tilde{d}_{il}(t) = \sqrt{a^2 + b^2 - 2ab\cos(\tilde{\theta}_{il}(t_0-\tau))} \qquad (2)$$

*where $a = (s + s'(t))$ is the distance traveled by the l-th PU during the temporal intervals $[t_0-\tau, t]$ and $\tilde{\theta}_{il}(t_0-\tau)$ is the estimated movement direction of the l-th PU towards the i-th CU at the time instant $(t_0-\tau)$.*

*Proof.* We refer the reader to [2], for the proof of Lemma 1. □

By means of Lemma 1, we can now derive the expression of the estimated CAP $\tilde{p}_{il}^{m}(t)$ when $t \in [t_0, (t_0 + \tau)]$ in the SPC scenario.

**Theorem 1.** *The estimated CAP $\tilde{p}_{il}^{m}(t)$ at time $t \in [t_0, (t_0+\tau)]$ assume the following expression:*

$$\tilde{p}_{il}^{m}(t) = \begin{cases} 1 & \textit{if } \tilde{d}_{il}(t) > R_{il} \\ P_{off}^{m} & \textit{otherwise} \end{cases} \quad \forall\, t \in [t_0, (t_0 + \tau)] \quad (3)$$

*Proof.* From the Lemma 1, we can estimate distance at time $t$. Utilizing $\tilde{d}_{il}(t)$, the theorem can proof directly. □

**Remark.** *The estimated CAP $\tilde{p}_{il}^{m}(t)$ depends on the estimated distance $\tilde{d}_{il}(t)$. Since it is assumed that the l-th PU does not change the velocity and direction during the interval $[t_0 - \tau, t_0 + \tau]$, the estimation procedure will encounter an error that depends on the PU mobility parameters and the temporal period $\tau$. The trade-off is discussed in the subsection IV-D.*

### *C. CAP estimation in Multiple PUs for Channel (MPC) scenario*

In this subsection, we derive (Theorem 2) the expression of the estimated CAP when $t \in [t_0, (t_0 + \tau)]$ in the MPC scenario, under the same assumptions of the Theorem 1.

**Theorem 2.** *If a number N of PUs, which are the elements of a primary user set $V^m$, use the same channel m simultaneously, then the estimated CAP $\tilde{p}_{iV}^{m}(t)$ at time $t \in [t_0, (t_0+\tau)]$ assume the following expression:*

$$\tilde{p}_{iV}^{m}(t) = \begin{cases} 1 & \textit{if } \tilde{d}_{il}(t) > R_{il} \quad \forall\, l \in V^m \\ P_{off}^{m} & \textit{otherwise} \end{cases} \quad \forall\, t \in [t_0, (t_0 + \tau)] \quad (4)$$

*Proof.* It is similar to the Theorem 1. □

**Remark.** *When the i-th CU is outside the protection range of all the N PUs belong to $V^m$ then the CAP is equal to one, otherwise it depends on the PU inactive probability. Since the probability that the i-th CU is inside the protection range of the arbitrary PU increases when N increases, then the CAP in the MPC scenario is lower than the SPC scenario.*

### *D. Trade-off between the PU position updating interval $\tau$ and distance estimation error*

It is worth noticing that the larger is $\tau$, the smaller is the updating rate of the PU position, i.e., the lower is the network overhead and energy consumption. However, the estimation of the distance for the next temporal period becomes less accurate. We explain this concept with an example, as shown in Figure. 5. Here, the non-dashed line is the exact PU movement pattern, and $v_l(t)$ and $\tilde{v}_l(t)$ represent the exact and estimated PU position at time $t$, respectively. Since the distance at time $t$ is estimated assuming that the PU does not change its velocity and direction during the interval $[(t_0 - \tau), (t_0 + \tau)]$, the estimation procedure will encounter an error when the PU changes these parameters during this interval. In particular, when $\tau$ increases, the error increases as well, and it has an impact on the accuracy of the estimation model which can be assessed in terms of Root Mean Square Error (RMSE). The RMSE increases with the increasing value of $\tau$. In particular, the RMSE increases when the network size decreases, since the smaller is the network size, the greater is the frequency that the PU change its direction, in according to the random waypoint mobility model [2]. In Section. VI, we will assess the impact of $\tau$ on the estimation of the CAP.

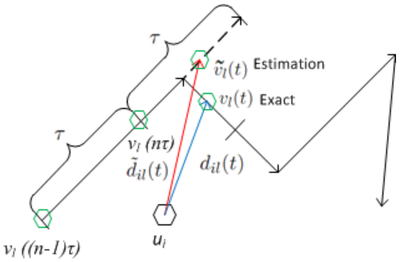


Fig. 5: PU movement pattern.

## V. Mobility-Aware Channel-Availability Based Channel Selection Technique

In this section, we discuss about proposed Mobility-aware Channel-Availability based channel Selection Technique. Based on the estimation model derived in the previous section, the CU selects the best channel in both the SPC (Theorem 3) and MPC (Theorem 4) scenarios with the highest value of the estimated CAP averaged over the next temporal period $[t_0, t_0 + \tau]$.

**Theorem 3.** *The expression of the MCAST in SPC scenario is the following:*

$$m_{opt}^{SPC} = \arg\max_{m} \tilde{q}_{il}^{m}(t_0, \tau) = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} \tilde{p}_{il}^{m}(t)dt \quad (5)$$

*where $\tilde{q}_{il}^{m}(t_0, \tau)$ denotes the estimated CAP averaged on $[t_0, t_0 + \tau]$ in the SPC scenario, that depends on the time instant $t_0$ and the period $\tau$.*

*Proof.* It follows by accounting for Theorem 1. □

**Theorem 4.** *The expression of the MCAST in MPC scenario is the following:*

$$m_{opt}^{MPC} = \arg\max_{m} \tilde{q}_{iV}^{m}(t_0, \tau) = \frac{1}{\tau} \int_{t_0}^{t_0+\tau} \tilde{p}_{iV}^{m}(t)dt \quad (6)$$

*where $\tilde{q}_{iV}^{m}(t_0, \tau)$ denotes the estimated CAP averaged on $[t_0, t_0 + \tau]$ in the MPC scenario, that depends on the time instant $t_0$ and the period $\tau$.*

*Proof.* It follows by accounting for Theorem 2. □

**Remark.** *The value of $\tilde{q}_{il}^{m}(t_0, \tau)$ and $\tilde{q}_{iV}^{m}(t_0, \tau)$ equal to one when the estimated distance during $[t_0, t_0 + \tau]$ is always greater then the protection range, it is equal to $P_{off}^{m}$ when the estimated distance is always less than or equal to the protection range during $[t_0, t_0 + \tau]$, while it is comprised between one and $P_{off}^{m}$ in the intermediate case.*

By exploiting the dynamic variation of the channel availability caused by the PU mobility, the proposed technique is able to outperform the static method that considers only the PU temporal activity. The simulation results in Section. VI highlight the benefits of using the proposed technique for selecting a channel in presence of PU mobility.

## VI. Simulation Results

In this section, first we evaluate via numerical experiments the performance of the proposed channel selection technique (MCAST). Then we prove its effectiveness by adopting MCAST in a routing metric, recently proposed in literature, referred to as OPERA [5].

### A. Performance evaluation

Figure. 6 (a-d) shows the performance comparison between the *mobility-aware method* (MCAST) and the *static method* in terms of maximum CAP ($\text{CAP}_{\text{max}}$), i.e., every $\tau$ seconds we consider the maximum CAP achievable from the best selected channel among the others, then we average it over the total number of periods considered in the simulation.

**Experiment 1:** It is plotted the exact and the estimated $\text{CAP}_{\text{max}}$ in the SPC scenario, along with the $\text{CAP}_{\text{max}}$ corresponding to the static method, versus the normalized PU protection range where $R_{il} = \{500m, 600m, 700m, ..., 1400m\}$. The adopted simulation set is defined as follows: the CU transmission range is $T_i = 100m$, CU interference range is $I_i = 200m$, the PU transmission range is $T_l = 300m$, the number of channels is $M = 5$, the PU inactive probability vector is $\{0.6, 0.2, 0.3, 0.5, 0.4\}$, the PU spectrum occupancy model is SPC, i.e., each channel is used by a single PU. The PUs move in a square region of side $a = 2000m$ according to the RWPM model, where the minimum velocity is $v_{min} = 5m/sec$ and maximum velocity is $v_{max} = 10m/sec.$

In Figure. 6 (a), we note that there is a very good agreement between the estimated and exact $\text{CAP}_{\text{max}}$ when $\tau = 10s$. The $\text{CAP}_{\text{max}}$ decreases in both methods when the PU protection range increases, and achieves the minimum value (given by the static method) when the normalized PU protection range is equal to one. This is reasonable because the greater is the protection range, the lower is the percentage of time in which the PUs are outside the protection range. For the static method, the $\text{CAP}_{\text{max}}$ is always 0.6 since it selects the channel according to the maximum PU inactive probability $P^m_{off}$.

In Figure. 6 (b), we note that the average error of the estimated $\text{CAP}_{\text{max}}$ increases by increasing $\tau$. This is because in the estimation model we assume that the PU does not change its velocity and direction during the interval $[(t_0-\tau), (t_0+\tau)]$. This error have an impact on the performance evaluation that means a trade-off between the effectiveness for the spectrum utilization and network overhead caused by the updating PU position mechanism.

**Experiment 2:** In this experiment, we consider the MPC scenario, i.e., each channel is used by multiple PUs, as shown in Figure. 6 (c, d). The adopted simulation set is the same defined in experiment 1, but we consider two PUs for each channel. We compare the $\text{CAP}_{\text{max}}$ in the SPC and MPC scenarios. Specifically, we note that the $\text{CAP}_{\text{max}}$ in the MPC scenario is lower than the SPC scenario. This is reasonable because, according to Theorem 2, the probability that the CU is inside the protection range of the PU increases when there are more PUs for each channel. The same considerations about the estimation model drawn for the SPC scenario are valid for the MPC scenario.

### B. Effectiveness

In this subsection, we evaluate the effectiveness of MCAST in a scenario of practical interest. Specifically, we adopt MCAST in a recently proposed routing metric designed for CRNs, referred to as OPERA, and analyze the network performance in terms of packet delay [5].

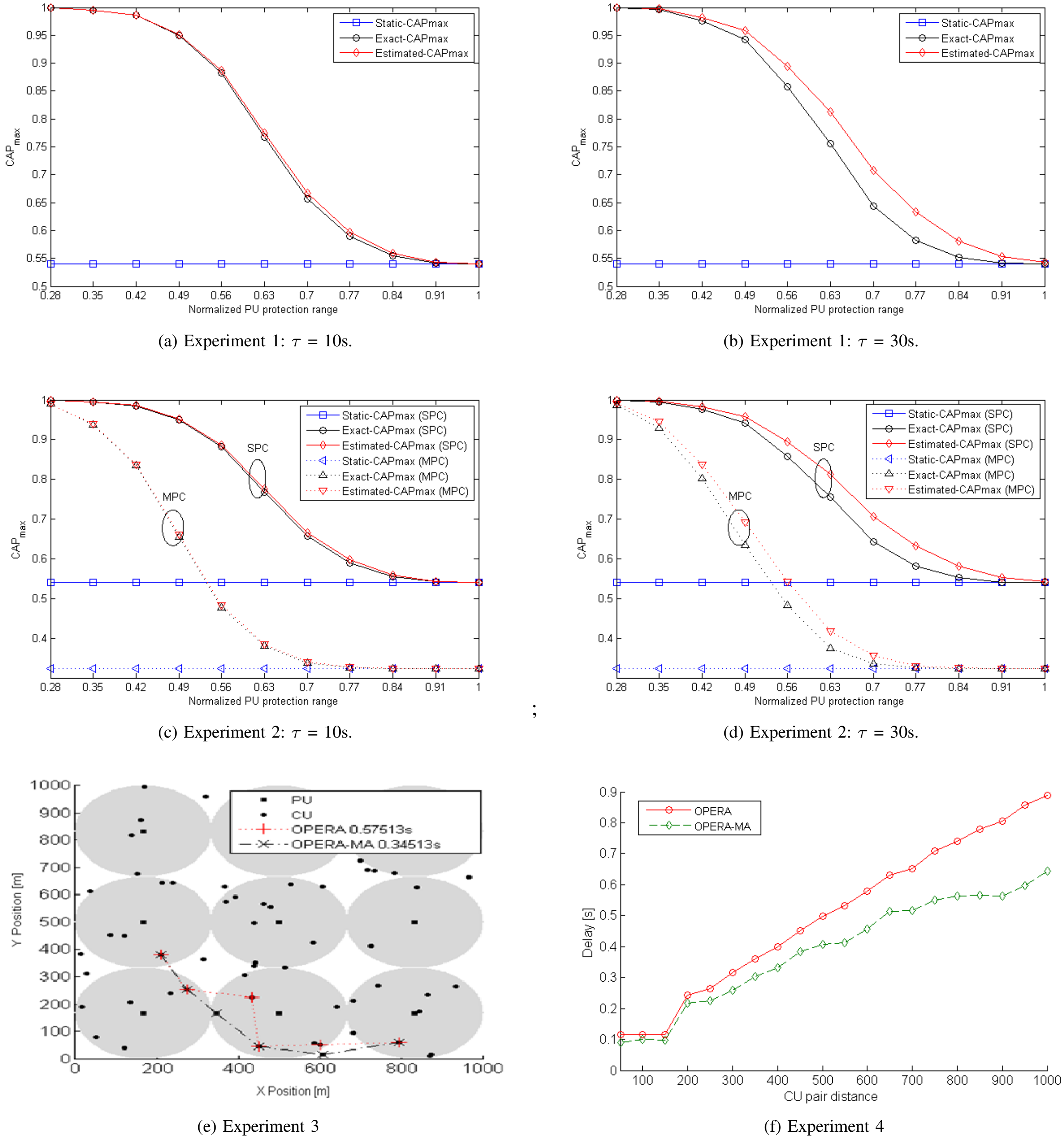
The network topology is shown in Figure. 6 (e) and it is similar to the one used in [1], with 64 CUs spread in a square region of side 1000m. The CU transmission standard is IEEE 802.11g, the packet length is L = 1500 bytes, the expected link throughput is $\bar{\psi}$ = 54Mbps, the transmission range of CU is equal to 200m, the transmission range of PU is equal to 166m and the number of channels is M = 2. Unlike the experiment in [1], we assume that the PUs are mobile and they are moving according to the RWPM.

**Experiment 3:** The experiment shows two different routes with the same source and destination, where the routes singled out by OPERA and OPERA with mobility-aware method (OPERA-MA), as shown in Figure. 6 (e). In the case of OPERA, where the static channel selection method is utilized, the delay is 0.57s. On the other hand, in the case of OPERA-MA we observe that the delay is significantly decreased to 0.34s, as it counteracts the adverse effect of PU mobility.

**Experiment 4:** In this experiment, we report the packet delay versus the distance between source and destination nodes for both the cases, as shown in Figure. 6 (f). First, we observe that the delay computed by both OPERA and OPERA-MA increases with the distance. This result is reasonable, because the longer is the path, the more is the number of PUs affecting it. However, we observe that OPERA-MA exhibits a significant improvement compared to OPERA when the distance increases, since more favorable paths are available by accounting for PU mobility.

## VII. Conclusion

In this paper, we proposed a novel mobility-aware channel-availability based channel selection technique for CRNs that ensures the selection of the channel with the highest CAP in a given temporal period. In fact, this technique takes advantage of the dynamic variation of channel-availability caused by the PU mobility and consequently outperforms the static method which is only based on the PU temporal activity. The numerical experiments corroborate the theoretical results. Moreover, we evaluate the effectiveness of MCAST in a scenario of practical interest by adopting this technique in a recently proposed routing metric designed for CRNs. The



(a) Experiment 1: $\tau$ = 10s.

(b) Experiment 1: $\tau$ = 30s.

(c) Experiment 2: $\tau$ = 10s.

(d) Experiment 2: $\tau$ = 30s.

(e) Experiment 3

(f) Experiment 4

Fig. 6: (a, b) Maximum CAP vs normalized PU protection range in SPC scenario; (c, d) Maximum CAP vs normalized PU protection range in both SPC vs MPC scenario; (e) Two different routes and the respective delays between the same pair source-destination, the routes singled out by OPERA and OPERA-MA; (f) Delay vs. CU pair distance for OPERA and OPERA-MA.

future research development foresee the design of a MAC protocol based on the proposed channel selection method.

## References

[1] K. R. Chowdhury and I. F. Akyildiz, ”CRP: a routing protocol for cognitive radio ad hoc networks”. IEEE Journal of Selected Areas in Communications (JSAC), Volume 29, Issue 4, 2011, pp. 794-804.

[2] A. S. Cacciapuoti, M. Caleffi, L. Paura, and Md. Arafatur Rahman, ”Channel availability for mobile cognitive radio networks”. Journal of Network and Computer Applications, Volume 47, 2014, pp. 131-136.

[3] S. C. Jha, U. Phuyal, M. M. Rashid, and V. K. Bhargava, ”Design of OMC-MAC: An Opportunistic Multi-Channel MAC with QoS Provisioning for Distributed Cognitive Radio Networks”. IEEE Transactions on Wireless Communications, Volume 10, 2011, pp. 3414-3425.

[4] D. Xue, E. Ekici, and X. Wang, ”Opportunistic Periodic MAC Protocol for Cognitive Radio Networks”. IEEE Globecom, 2010, pp. 1-6.

[5] M. Caleffi, I. F. Akyildiz, and L. Paura, ”Opera: Optimal routing metric for cognitive radio ad hoc networks”. IEEE Transactions on Wireless Communications, Volume 11, Issue 8, 2012, pp. 2884-2894.

[6] S. Yangand, F. Yuguang, and Z. Yanchao ”Stochastic Channel Selection in Cognitive Radio Networks”. IEEE Globecom 2007, 2007, pp. 4878-4882.

[7] H. N. Pham, J Xiang, Y. Zhang, and T. Skeie, ”QoS-Aware Channel Selection in Cognitive Radio Networks: A Game-Theoretic Approach”. IEEE Globecom, 2008, pp. 1-7.

[8] Y. Yong, S. R. Ngoga, D. Erman, and A. Popescu, ”Competition-based channel selection for cognitive radio networks”. IEEE Wireless Communications and Networking Conference 2012, 2012, pp. 1432-1437.

[9] T. Camp, J. Boleng, and V. Davies, ”A survey of mobility models for ad hoc network research”. Wireless Communications and Mobile Computing, Volume 2, 2002, pp. 483-502.

[10] Y. Chen, Q. Zhao, and A. Swami, ”Joint design and separation principle for opportunistic spectrum access”. IEEE Asilomar Conference on Signals Systems and Computers, 2006, pp. 696-700.

[11] A. S. Cacciapuoti, I. F. Akyildiz, and L. Paura, ”Primary-User Mobility Impact on Spectrum Sensing in Cognitive Radio Networks”. Proc. of IEEE Symposium on Personal, Indoor, Mobile and Radio Communications (PIMRC 2011), Toronto, Canada, 2011, pp. 451-456.

[12] A. W. Min, X. Zhang, and K. G. Shin, ”Detection of Small-Scale Primary Users in Cognitive Radio Networks”. IEEE Journal on Selected Areas in Communications, Volume 29, 2011, pp. 349-361.

[13] L. Xiao, L. J. Greenstein, and N. B. Mandayam, ”Sensor-assisted localization in cellular systems”. IEEE Trans. Wireless Commun, Volume 6, 2007, pp. 4244-4248.

[14] S. Liu, Y. Chen, W. Trappe, and L. J. Greenstein, ”Non-interactive localization of cognitive radios based on dynamic signal strength mapping”. IEEE/IFIP WONS, 2009, pp. 85-92.

# Computer-aided Knowledge Extraction and Management for Decision Supporting Processes

Lidia Ogiela
AGH University of Science and Technology
Cryptography and Cognitive Informatics Research Group
Krakow, Poland
e-mail: logiela@agh.edu.pl

Marek R. Ogiela
AGH University of Science and Technology
Cryptography and Cognitive Informatics Research Group
Krakow, Poland
e-mail: mogiela@agh.edu.pl

*Abstract*—**This publication presents the essence of the process of obtaining knowledge for the purpose of managing data in strategic decision-taking processes. A method of analysing selected datasets will be described by reference to the semantic analysis and interpretation of data. A new class of systems supporting strategic information management processes – Understanding Based Management Financial Leverage Ratios Support Systems (UBMFLRSS) – has been chosen for the analysis. These systems are designed for cognitively analysing financial leverage ratios (financial debt ratios) and reasoning about sources financing the company's assets and the proportion of external capital based on analyses of short-term and long-term liabilities, as well as about the effectiveness of expenditure and the interest paid. Based on the semantic analysis of the value of leverage ratios, it is possible to assess the current standing of the enterprise and its future situation by indicating the direction of change that should be made.**

***Keywords-cognitive financial systems; knowledge extraction; data mining; semantic interpretation***

## I. INTRODUCTION

The authors of this paper have been developing cognitive systems for the semantic analysis of various data for many years. These have been discussed, among others, in the following publications [5], [8], [9], [10], [11], [12].

Cognitive data analysis systems have been divided into various classes according to the type of data they analyse [3], [10]. Thus, the following classes of systems were developed: decision-making, image data analysis, signal data analysis, personal data analysis, automatic control and management process support [5], [10], [11].

The subject of this paper is to discuss the class of systems supporting strategic information management processes [2], [4], [13].

The class of cognitive data analysis systems which support management processes has been split into four main subclasses. The essence of this split is the interpretation of groups of financial ratios which influence (which bring about) the current standing of the enterprise.

Four main system classes have been distinguished in the group of systems supporting financial data management processes [10]:

- Cognitive Understanding Based Management Liquidity Ratios Support Systems (UBMLRSS) – systems for analysing enterprise liquidity ratios, which reason about the amount and the solvency of the working capital of the company as well as about the company's current operations;
- Cognitive Understanding Based Management Activity Ratios Support Systems (UBMARSS) – systems for analysing turnover ratios, which reason about how fast assets rotate and how productive they are;
- Cognitive Understanding Based Management Profitability Ratios Support Systems (UBMPRSS) – systems for analysing profitability ratios, which reason about the financial efficiency of the business of a given unit based on the relationship between the financial results, the sales of goods and services and the cost of sales, and also
- Cognitive Understanding Based Management Financial Leverage Ratios Support Systems (UBMFLRSS) – systems for analysing financial leverage ratios (financial debt ratios), which reason about the sources financing the company's assets and the proportion of external capital by analysing short-term and long-term liabilities; they also reason about the effectiveness of expenditure and the interest paid.

The classes of systems for analysing enterprise liquidity ratios, turnover ratios and profitability ratios have already been analysed and discussed, among others, in publications [9], [10].

The class of systems for analysing debt ratios has not been discussed yet and this is why it is the main subject of this publication. This is an innovative solution that supports cognitive data analysis processes of financial leverage ratios and processes for supporting enterprise financial figure management based on an analysis of debt ratios.

The purpose of this study is to present algorithms for semantic data analysis in the group of UBMFLRSS systems which are used to support strategic decision-taking in management processes.

Semantic analysis processes are used to interpret various sets of data/information, but this publication will only discuss semantic analysis processes dedicated to supporting information management processes. The semantic analysis conducted using UBMFLRSS systems allows the current standing of the enterprise to be identified and also shows what decisions should be taken in the future to improve the current situation or maintain it (if the standing of the enterprise is very good). This kind of analysis should be conducted not only for a selected enterprise, but should also identify the impact of the external environment. This is why systems for the semantic analysis of data used for analysis and supporting enterprise management, concentrate their action around [9]:

- Analysis of the internal situation of the company;
- Analysis of the external situation of the company;
- Predicting the future situation;
- Improving decision-making processes;
- Support strategic decision-making;
- Support enterprise management processes;
- Support enterprise management processes in the global aspects.

This situation is shown in Figure 1.

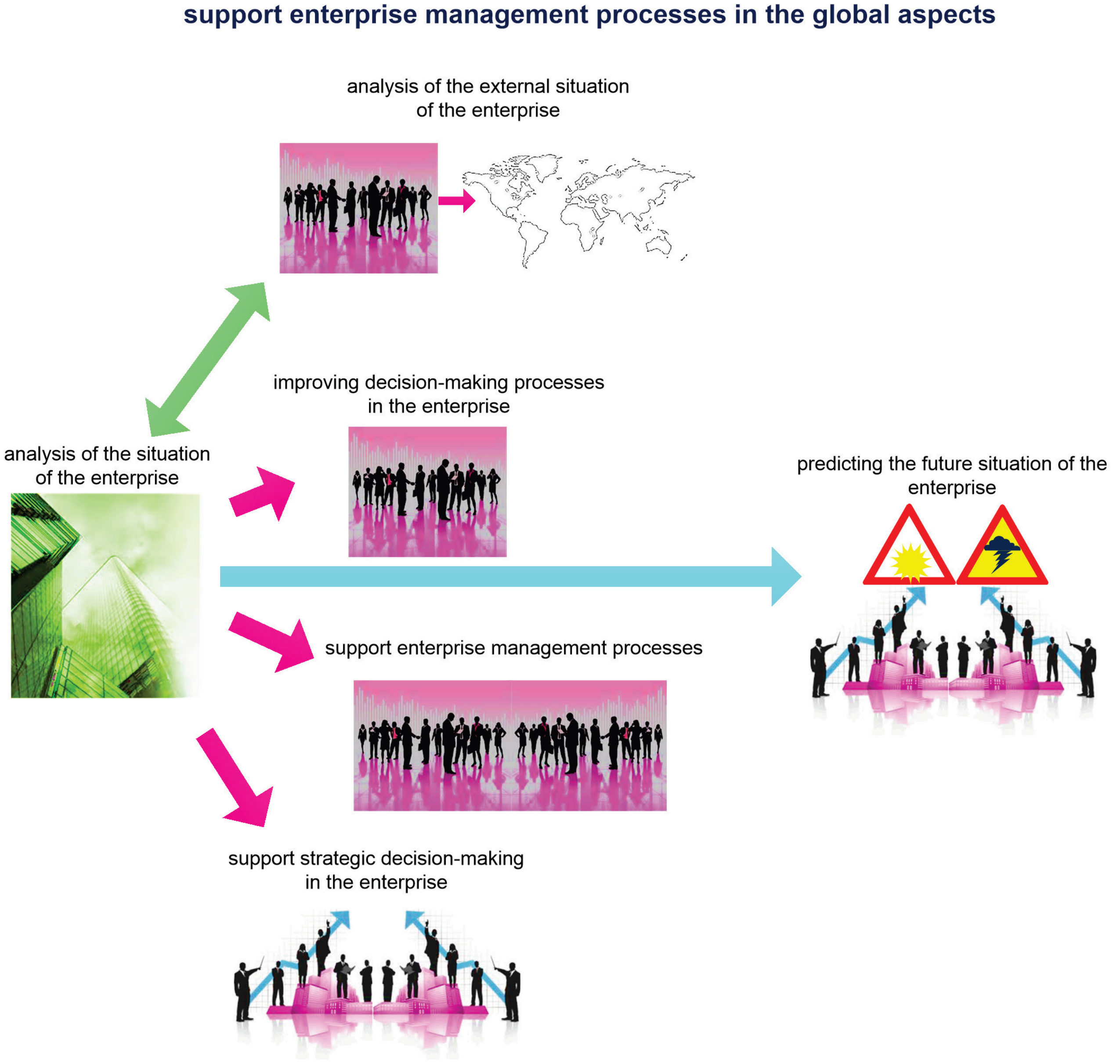

support enterprise management processes in the global aspects
analysis of the external situation of the enterprise
improving decision-making processes in the enterprise
analysis of the situation of the enterprise
predicting the future situation of the enterprise
support enterprise management processes
support strategic decision-making in the enterprise

Figure 1. The enterprise management processes in the global aspects.

The analysed datasets and the standing of the enterprise can be cognitively analysed by assessing various datasets. This publication describes an analysis of ratios used to assess the level of debt of a given enterprise.

In Section 2, we will present three sub-classes of UBMFLRSS systems as examples of intelligent enterprise debt analysis systems – the UBMFLRSS-G$_{(td\text{-}ld)}$ systems, the UBMFLRSS-G$_{(td\text{-}ls)}$ systems, and the UBMFLRSS-G$_{(dsc\text{-}ic)}$

systems. Section 3 concludes the paper with a summary of the conducted research.

## II. UBMFLRSS AS AN EXAMPLE OF ENTERPRISE DEBT ANALYSIS SYSTEMS

Enterprise debt applies to a situation in which the enterprise uses any form of financial support and not just exclusively its own capital. It applies to any situation in which the enterprise users external capital. In this situation, the enterprise is indebted.

The assessment of the enterprise debt situation is aimed at determining the extent to which the enterprise finances itself with its own funds and to which it is financed with funds coming from outside. In this context, it is possible to assess the proportion of own capital to external capital in financing the enterprise. The most important element in assessing the debt situation is to determine the impact of external capital on enterprise operations and the degree to which the financial independence of the enterprise is at risk [1]. This type of assessment also results in identifying the costs of using external capital and the cost-effectiveness of this solution.

The values of the following debt ratios can be analysed:

- Debt level ratios;
- The company’s ability to service its debt.

The following ratios are distinguished in the group of ratios identifying the debt level [1]:

- Total debt ratio;
- Long-term debt;
- Debt to equity;
- Long-term debt to equity;
- Liability structure;
- Long-term liability coverage with net fixed assets;
- Interest coverage.

The following ratios are distinguished in the group of ratios identifying the ability of the enterprise to service its debt:

- Debt service coverage;
- Interest coverage;
- Debt service coverage with the cash surplus.

Semantic data analysis systems are used to assess the current situation of the enterprise based on the semantic interpretation of a selected group of ratio data [7], [9].

In cognitive financial systems, the following may be analysed [9], [10]:

- The financial situation of enterprise – financial ratios;
- The economic situation of enterprise;
- The surroundings of enterprise;
- The situation and condition of:
  - customers,
  - providers,
  - others companies,
- And the influence of the environment of the company.

In cognitive systems for the semantic analysis of data, namely debt ratios, it is possible to determine:

- The degree to which enterprise operations are financed with its own funds;
- The degree to which enterprise operations are financed with external funds – the enterprise's debt;
- The proportion of own capital to external capital in corporate finance;
- The debt situation – by determining the impact of external capital on enterprise operations and the degree to which the financial independence of the enterprise is at risk;
- The costs and profitability of using external capital for enterprise operations.

Computer-aided cognitive enterprise management systems analyse various types of financial ratios. Based on the situation of the enterprise, the next part of analysis is the selection of type financial ratios that will be important and analysed [10].

After this, a proper class of cognitive financial systems should be selected, based on the type of analysed indicators [1].

In this systems, it is necessary to define a formal grammar to analysed selected data and indicators for the semantic features of data sets [5], [6], [8], [11], [12], [13], [14].

The next part of analysis and computer-aided enterprises management [7] is the evaluation of the enterprise situation, based on the important financial ratios. This analysis is especially important for [10]:

- Describing the present situation of the enterprise;
- Supporting enterprise management processes;
- Understanding of the current state of companies;
- Understanding of the causes of the current situation of the company;
- Describing the future situation of the enterprise.

The process of data mining in computer-aided enterprises management systems is shown in Figure 2.

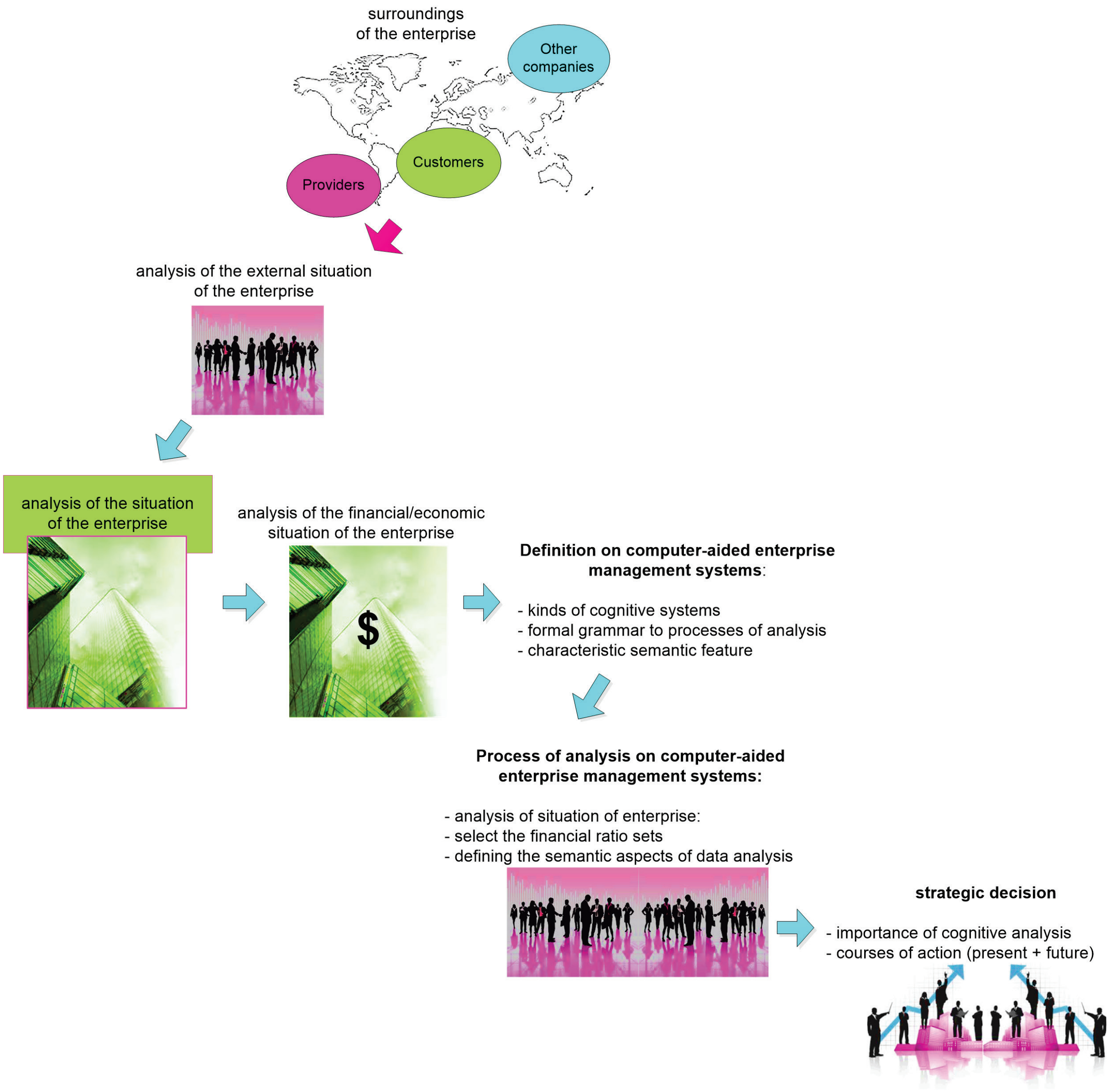

surroundings
of the enterprise
Other
companies
Customers
Providers
analysis of the external situation
of the enterprise
analysis of the situation
of the enterprise
analysis of the financial/economic
situation of the enterprise
$
**Definition on computer-aided enterprise management systems**:
- kinds of cognitive systems
- formal grammar to processes of analysis
- characteristic semantic feature
**Process of analysis on computer-aided enterprise management systems:**
- analysis of situation of enterprise:
- select the financial ratio sets
- defining the semantic aspects of data analysis
**strategic decision**
- importance of cognitive analysis
- courses of action (present + future)

Figure 2. The process of data mining in computer-aided enterprise management systems.

This paper will discuss UBMFLRSS systems designed for the semantic analysis of enterprise debt.

In the financial cognitive UBMFLRSS systems, supporting enterprises management, authors proposed a sequential grammar [9], [10], [11] as a formal definition.

In this paper, the semantic analysis procedures will be adopted for enterprise management and strategic-decision making tasks using cognitive information systems – example UBMFLRSS.

### A. *The UBMFLRSS-G$_{(td\text{-}ld)}$ Systems*

The following ratios have been adopted to analyse enterprise standing in UBMFLRSS systems as described after interpreting selected corporate debt ratios:

- total debt ratio,
- long-term debt ratio.

In UBMFLRSS-G$_{(td\text{-}ld)}$ systems for the semantic analysis of the debt value, the following symbols have been introduced:

- $v_{td}$ – the value of the total debt ratio,
- $v_{ld}$ – the value of the long-term debt ratio.

A formal grammar definition in these systems has the following form:

$$G_{(td\text{-}ld)} = (V_{N(td\text{-}ld)}, V_{T(td\text{-}ld)}, P_{(td\text{-}ld)}, S_{(td\text{-}ld)}) \qquad (1)$$

where:

$V_{N(td\text{-}ld)}$ = {DEBT1, HIGH_DEBT1, OPTIMAL_DEBT1, LOW_DEBT1} – the set of non-terminal symbols,

$V_{T(td\text{-}ld)}$ = {a, b, c}– the set of terminal symbols,

where:

a ∈ [0; 0,57), b ∈ [0,57; 0,67], c ∈ (0,67; 1]
$S_{(td\text{-}ld)} \in V_{N(td\text{-}ld)}$,
$S_{(td\text{-}ld)}$ = DEBT1,
$P_{(td\text{-}ld)}$ – set of productions:

1. DEBT1 →HIGH_DEBT1 | OPTIMAL_DEBT1 | LOW_DEBT1 |
2. HIGH_DEBT1 → BC | CB | CC
3. OPTIMAL_DEBT1 → BB
4. LOW_DEBT1 → AA | AB | AC | CA | CB
5. A → a
6. B → b
7. C → c.

### B. The UBMFLRSS-$G_{(td\text{-}ls)}$ Systems

The second example of UBMFLRSS systems for the semantic analysis of data described by interpreting selected enterprise debt ratios was created for analysing the following ratios:

- total debt ratio,
- liability structure.

In UBMFLRSS-$G_{(td\text{-}ls)}$ systems for the semantic analysis of the debt value, the following symbols have been introduced:

- $v_{td}$ – the value of the total debt ratio,
- $v_{ls}$ – the value of the liability structure ratio.

A formal grammar definition in these systems has the following form:

$$G_{(td\text{-}ls)} = (V_{N(td\text{-}ls)}, V_{T(td\text{-}ls)}, P_{(td\text{-}ls)}, S_{(td\text{-}ls)}) \quad (2)$$

where:
$V_{N(td\text{-}ls)}$ = {DEBT2, HIGH_DEBT2, OPTIMAL_DEBT2, LOW_DEBT2} – the set of non-terminal symbols,
$V_{T(td\text{-}ls)}$ = {a, b, c}– the set of terminal symbols,
where:
a ∈ [0; 0,57), b ∈ [0,57; 0,67], c ∈ (0,67; 1]
$S_{(td\text{-}ls)} \in V_{N(td\text{-}ls)}$,
$S_{(td\text{-}ls)}$ = DEBT2,
$P_{(td\text{-}ls)}$ – set of productions:

1. DEBT2 →HIGH_DEBT2 | OPTIMAL_DEBT2 | LOW_DEBT2 |
2. HIGH_DEBT2 → CC
3. OPTIMAL_DEBT2→ BB | BC | CB
4. LOW_DEBT2 → AA | AB | AC | BA | CA
5. A → a
6. B → b
7. C → c.

### C. The UBMFLRSS-$G_{(dsc\text{-}ic)}$ Systems

The third example of UBMFLRSS systems for the semantic analysis of data described by interpreting selected enterprise debt ratios designed for assessing the enterprise ability to service its debts was created for analysing the following ratios:

- debt service coverage,
- interest coverage.

In UBMFLRSS-$G_{(dsc\text{-}ic)}$ systems for the semantic analysis of the company's ability to service its debt, the following symbols have been introduced:

- $v_{dsc}$ – the value of the debt service coverage ratio,
- $v_{ic}$ – the value of the interest coverage ratio.

A formal grammar definition in these systems has the following form:

$$G_{(dsc\text{-}ic)} = (V_{N(dsc\text{-}ic)}, V_{T(dsc\text{-}ic)}, P_{(dsc\text{-}ic)}, S_{(dsc\text{-}ic)}) \quad (3)$$

where:
$V_{N(dsc\text{-}ic)}$ = {DEBT_SERVICE1, HIGH_DEBTSERVICE1, MEDIUM_DEBTSERVICE1, LOW_DEBTSERVICE1} – the set of non-terminal symbols,
$V_{T(dsc\text{-}ic)}$ = {a, b, c}– the set of terminal symbols,
where:
a ∈ [0; 1), b ∈ [1; 1,5], c ∈ (1,5; 2,5]
$S_{(dsc\text{-}ic)} \in V_{N(dsc\text{-}ic)}$,
$S_{(dsc\text{-}ic)}$ = DEBT_SERVICE1,
$P_{(dsc\text{-}ic)}$ – set of productions:

1. DEBT_SERVICE1 →HIGT_DEBTSERVICE1 | MEDIUM_DEBTSERVICE1 |LOW_DEBTSERVICE1
2. HIGH_DEBTSERVICE1 → CC
3. MEDIUM_DEBTSERVICE1 → BB | BC | CB
4. LOW_ DEBTSERVICE1 → AA | AB | BA | AC | CA
5. A → a
6. B → b
7. C → c.

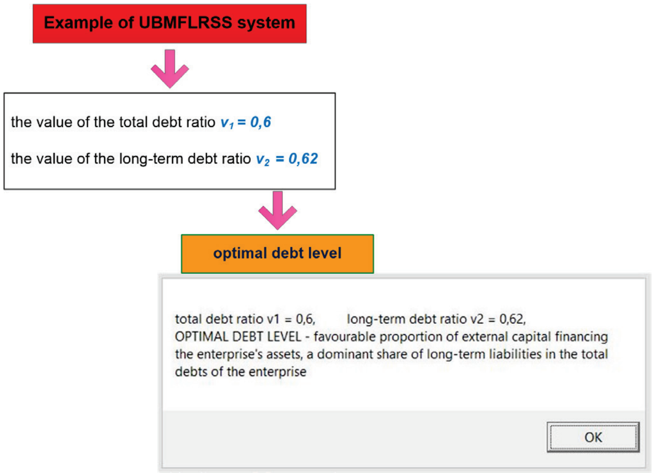
The three examples of systems for the semantic analysis of financial data presented above are designed for assessing the debt level of an enterprise and the ability of this enterprise to service its debt.

Three independent linguistic formalisms in the form of sequential grammars, each for a separate subclass of UBMFLRSS, have been proposed for the semantic analysis of financial data.

The semantic analysis conducted in UBMFLRSS allows the debt situation of the enterprise to be assessed and indicates its possible consequences. In this sense, it can be used to support information management. What is supported in the described case is the management of financial figures describing the debt situation of the enterprise.

Examples of cognitive data analysis systems illustrate systems that support decision making based on extracted knowledge elements. In UBMFLRSS systems for the semantic analysis of data, the knowledge means information about the financial standing, and particularly the debt of the enterprise as well as its ability to service the debt contracted. Extracting knowledge elements in UBMFLRSS systems is fund components of information sets, which components will be used directly to assess the situation of the enterprise. Extracting significant features that undergo the semantic analysis process helps to more effectively manage the information held and adds the elements of the semantic interpretation of financial information to the processes of managing this information.

An example of the operation of a UBMFLRSS system is shown in Figure 3.


Example of UBMFLRSS system
the value of the total debt ratio $v_1 = 0,6$
the value of the long-term debt ratio $v_2 = 0,62$
optimal debt level
OK

total debt ratio v1 = 0,6,  long-term debt ratio v2 = 0,62,
OPTIMAL DEBT LEVEL - favourable proportion of external capital financing the enterprise's assets, a dominant share of long-term liabilities in the total debts of the enterprise

Figure 3. Example of UBMFLRSS-$G_{(td-ld)}$ cognitive financial systems.

The example of the UBMFLRSS system presented here is used to analyse ratio data, namely selected groups of debt ratios: the total debt ratio and the long term debt ratio.

Based on this semantic analysis, the system interprets the situation of the enterprise as good from the perspective of its debt level. The semantic interpretation is to determine the optimal level of debt understood as a favourable proportion of external capital used to finance the company's assets and its operations, as well as a dominant share of long term liabilities in the total debt of the enterprise.

Based on the analysis of selected financial ratios the situation of the company is described. The example of cognitive analysis of financial ratios shows how the two selected ratios shape the situation of the enterprise.

## III. Conclusions and Future Work

Cognitive data analysis systems are used for semantic analysis which consists in extracting semantic information from analysed datasets and interpreting this information. Thus, cognitive analysis processes ensure that data/information will be analysed with regards to its meaning. Linguistic formalisms in the form of definitions of formal grammars (sequential, tree or graph) are used to describe the analysed data sets. Financial information management systems are an example of cognitive data analysis systems. In this class of systems, linguistic formalisms in the form of sequential grammars have been proposed for the formal description of the analysed data.

The analysis of ratio data in cognitive systems was proposed for the purpose of improving processes of strategic (financial) information management. This improvement is possible because linguistic formalisms are used in the process of analysing financial data. This not only allows the analysed financial values to be interpreted, but it also helps assess the situation of the enterprise and indicates the possible directions of its change if necessary or it shows that there is no need to implement any remedial action. In this publication, we proposed a new approach to the semantic analysis of financial data. Evaluation of the effectiveness of the proposed solutions will be the subject of future work.

## Acknowledgment

This work has been supported by the National Science Centre, Republic of Poland, under project number DEC-2012/05/B/HS4/03625.

## References

[1] L. Bernstein, J. Wild, “Analysis of Financial Statements”, McGraw-Hill, New York, 2000.

[2] S. Buchanan, D. McMenemy, „Digital service analysis and design: The role of process modelling”, International Journal of Information Management, vol. 32(3), 2012, pp. 251–256.

[3] H. Cohen, C. Lefebvre (Eds.), “Handbook of Categorization in Cognitive Science,” Elsevier, The Netherlands, 2005.

[4] S. Grossberg, “Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world”, Neural Networks, vol. 37, 2012, pp. 1-47.

[5] T. Hachaj, M.R. Ogiela, “Framework for cognitive analysis of dynamic perfusion computed tomography with visualization of large volumetric data”, Journal of Electronic Imaging, vol. 21(4), 2012, paper No: 043017, doi: 10.1117/1.JEI.21.4.043017

[6] A. Kornai, “Mathematical Linguistics,” Springer Verlag, Berlin Heidelberg, 2008.

[7] K.C. Laudon, J.P. Laudon, “Management Information Systems – Managing the Digital Firm,” Seventh Edition, Prentice-Hall International: Inc., 2002.

[8] L. Ogiela, “Syntactic Approach to Cognitive Interpretation of Medical Patterns,” in: C. Xiong at all (Eds.), Intelligent Robotics and Applications, First International Conference, ICIRA 2008, Wuhan, China, 15-17 October 2008, LNAI 5314, Springer-Verlag Berlin Heidelberg, 2008, pp. 456-462.

[9] L. Ogiela, “Towards Cognitive Economy”, Soft Copmputing, vol. 18 (9), 2014, pp. 1675-1683

[10] L. Ogiela, “Data management in cognitive financial systems,” International Journal of Information Management, vol. 33, 2013, pp. 263-270.

[11] L. Ogiela, M.R. Ogiela, “Semantic Analysis Processes in Advanced Pattern Understanding Systems”, in: T.H. Kim at all (Eds.), Advanced Computer Science and Information Technology, 3rd International Conference on Advanced Science and Technology AST 2011, Jeju Island, South Korea, 15-17 Jun 2011, Communications in Computer and Information Science, vol. 195, 2011, pp. 26-30.

[12] M.R. Ogiela, U. Ogiela, “The use of mathematical linguistic methods in creating secret sharing threshold algorithms”, Computers & Mathematics with Applications, vol. 60(2), 2010, pp. 267-271.

[13] P. TalebiFard, V. C. M. Leung, “Context-Aware Mobility Management in Heterogeneous Network Environments”, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, vol. 2(2), 2011, pp. 19-32.

[14] S.C. Zhu, D. Mumford, “A stochastic grammar of images,” Foundations and Trends in Computer Graphics and Vision, vol. 2(4), 2006, pp. 259-362.

# Unified POF Programming for Diversified SDN Data Plane Devices

Haoyu Song, Jun Gong, Hongfei Chen, Justin Dustzadeh

Huawei Technologies
Santa Clara, CA, USA, 95124
email: {haoyu.song, jun.gong, hongfei.chen, justin.dustzadeh}@huawei.com

***Abstract*—Software Defined Networking (SDN) will ultimately evolve to be able to program the network devices with customized forwarding applications. The ability to uniformly program heterogeneous forwarding elements built with different chips is desirable. In this paper, we discuss a data plane programming framework suitable for a flexible and protocol-oblivious data plane and show how OpenFlow can evolve to provide a generic interface for platform-independent programming and platform-specific compiling. We also show how an abstract instruction set can play a pivotal role to support different programming styles which map to different forwarding chip architectures. As an example, we compare the compiler-mode and interpreter-mode implementations for a Network Processing Unit (NPU) based forwarding element and conclude that the compiler-mode implementation can achieve a performance similar to that of a conventional non-SDN implementation. Built upon our Protocol-Oblivious Forwarding (POF) vision, this work presents our continuous efforts to complete the ecosystem and pave the SDN evolving path. The programming framework could be considered as a proposal for the OpenFlow 2.0 standard.**

***Keywords–SDN; OpenFlow; POF; data plane; programming.***

## I. INTRODUCTION

It has been envisioned that in SDN the network intelligence should be moved to software as much as possible in order to support agile, flexible, and low-cost network service deployments. Programmable Forwarding Elements (FE) are essential to enable this vision and represent a big leap from the current network device application paradigm. These FEs will be shipped as white-box just like bare-metal servers without fixed functions or pre-installed applications. User can program the device through a standard-based open interface. Therefore, the future SDN operation can be modeled as follows. First, the user determines the entire forwarding protocols and behavior through device-level programming. Note that this is done through high level programming over high level device abstractions. After this step, the white-box is equipped with customized functions tailored for operator needs. Then, the operator applies runtime control to operate these devices through network-level and service-level programming. Any third party can produce software to program and configure the programmable FEs. Any third party can also produce network-level application software which taps into the customized FEs to offer various network services. Depending on the actual use cases, the role of device programmer, service programmer, and network user can be overlapped or independent. The advantages of this network operation model are obvious. Network applications can be programmed on-the-fly and deployed in real time. Service innovation is never so easy and accessible before. Moreover, the system time-to-market can be significantly reduced and the life cycle of FEs greatly extended.

In the arena of programmable data plane devices, Central Processing Unit (CPU) and NPU-based FEs are clearly qualified candidates, but so far, there is lack of an open and standard interface for forwarding application programming on these devices. In most cases, these devices are still programmed by vendors and shipped to users in the form of virtual or physical appliance. Some open-source soft switches, such as Open Virtual Switch (OVS) [1], allow user to modify its behavior but apparently this process is labor intensive and target dependent. Hence, the application model of these FEs is not so much different from those built with fixed-function switch chips based on Application Specific Integrated Circuit (ASIC).

While not fully programmable, ASIC chips are usually configurable to some extent and able to handle most of popular Data Center (DC) switch applications. ASIC-based FEs can be considered to have pre-installed packages or standard library functions. With certain negotiation process, such as Table Type Pattern (TTP) Negotiable Datapath Model (NDM) [2], ASIC-based FEs can still be controlled under the same SDN framework, as if they were programmed by the controller.

Recently, a new breed of SDN-optimized programmable chip is investigated [3]–[5]. These chips aim to support flexible SDN application programming without compromising performance. If succeed, this new contender will further accelerate the SDN transformation. It is worth to mention that Field Programmable Gate Array (FPGA), a reconfigurable chip by nature, can also potentially play a similar role with proper design-flow refactoring. SDNet [6] represents such an effort.

For the foreseeable future, diverse FEs built with different chips will coexist in various network segments. As such, it is critical to have a unified framework, not only to control and program these FEs, but also to hide the heterogeneous substrate architecture and present a unified programming interface to SDN controller and applications. We position OpenFlow [7] as the center pillar of this framework. OpenFlow abstracts the SDN data path as a pipeline of tables and actions. This model is arguably the easiest way to map the forwarding functions to any target FEs. However, further investigation and work are needed to address some of the challenges with the current approach (e.g., fixed protocol support and stateless data path) [5], [8].

We believe the next generation of OpenFlow (e.g., OpenFlow 2.0) should offer the following capabilities: (1) Allow a protocol-oblivious data plane so that no packet format and network behavior need to be hard-coded in FEs. This capability is important to maximize SDN's flexibility and extensibility. (2) Allow an FE-agnostic SDN controller so that the data plane abstraction can help isolate the controller from the FE implementation details. This capability is important to minimize SDN's efforts to program heterogeneous substrate platforms. (3) Allow coexistence of static programming through the use of packages or library functions and dynamic runtime programming/reconfiguration through the use of flow instructions. This capability extends the usability of diversified FEs and can offer the needed flexibility to satisfy some special SDN use-case requirements. While audacious, these goals represent the right direction for OpenFlow evolution. In this paper, we present an OpenFlow-based SDN FE programming framework and provide our experience on realizing it.

The remainder of the paper is structured as follows: Section II describes the proposed programming framework; Section III provides a case study on an NPU-based platform; Section IV discusses the related work; and Section V concludes the paper.

## II. Unified Programming Framework

The unified SDN data-plane FE programming framework is depicted in Figure 1. The center pillar of this framework is the standardized OpenFlow 2.0 interface, which provides a set of generic instructions, as well as other data-plane provision and monitoring mechanisms. The interface provides a decoupling point between the control plane and the data plane. It is versatile and protocol/platform-agnostic [8].

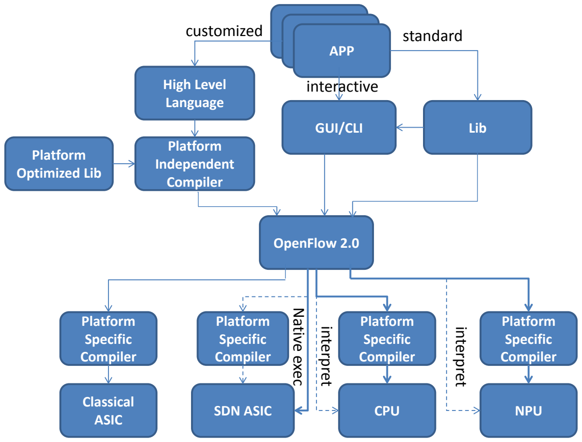


Figure 1: SDN Data Plane Programming Framework

### A. Intermediate OpenFlow Interface

OpenFlow is pivotal for the data plane programmability. The key to a successful design of such a programming interface is to make it work at the right abstraction levels. In particular, the interface should not be tied to a particular FE architecture. Instead, it should allow programs to be easily mapped to any target while allowing specific optimizations to fully exploit the target-specific capability.

To this end, we propose a simple yet generic abstract forwarding model, as shown in Figure 2. In this model, the "In Ports" and "Out Ports" are the source and sink of a packet under processing. The port can be either physical or logical. It can be anything that is out of the scope of the processing directly programmed by user. For example, the controller, a service card, some black-box network functions, and packet recirculation can all be abstracted as ports. This abstraction guarantees only a single packet is in the processing pipeline at a time and the packet is uniquely identified by its input port.

The packet processing is abstracted as a sequence of search tables and the corresponding actions triggered by table lookup results. Note that this view is also roughly held by OpenFlow but we purify it to an extreme. The table can match on any designated packet field or metadata, and the matching result points to an action which comprises a block of instructions.

The abstract forwarding model can easily describe any packet processing tasks. For example, to map to some target hardware with a front-end parser, the very initial table is defined as a port table and the following action contains instructions which parse packets and extract header fields. The number of instructions allowed in an action depends on the target hardware and the performance constraints. A target-specific compiler can also find the parallelism opportunities within an action and takes advantage of it.

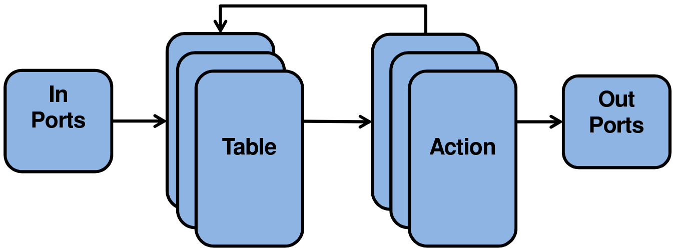


Figure 2: Abstract Forwarding Model

The core of our proposed OpenFlow 2.0 interface is a set of generic "flow" instructions (i.e., POF Flow Instruction Set (FIS)) [9], which are used to program the actions. These instructions function as the intermediate language between the platform-independent programming environment and each individual target platform. The instructions are grouped and summarized as follows:

- *Packet/Metadata editing:* set field, add field, delete field, math/logic operations on field in packet or metadata
- *Flow Metadata manipulation:* read, write global data (e.g., counters, meters) for stateful operations
- *Algorithm/Function procedure:* checksum, hashing, random number generating, etc. Extensible to include other standalone black-box functions
- *Table access:* go to table (non return), search table (return to calling instruction) with keys extracted from packet and metadata
- *Output:* to physical/virtual/logical port, with sampled/data-path generated packets, or original/modified/mirrored/cloned packets
- *Jump/Branching:* conditional and unconditional, absolute and relative
- *Active data path:* insert/delete/modify flow entry, insert/delete flow table
- *Event:* timers

These instructions operate on the objects such as packet data, meta data, and flow tables. Note that the instruction set we defined by no means complete. More instructions can be included in the future as the scope of packet processing is extended to cover tasks such as queuing and scheduling.

In addition to making the flow instructions protocol-oblivious, we propose other new features to enhance the programmability and to enable performance optimization. One notable addition is the ability to abstract the actions associated with each flow entry as a piece of program. This provides several advantages. For example, it allows decoupling match keys and actions. The actions for flow entries, in form of instruction blocks, can be downloaded to FEs separately from flow entry installations. When each instruction block is assigned a unique identifier (ID), the flow entry only needs to include a block ID to infer the associated actions. By doing this, not only different flow entries can share the same instruction block while an instruction block is only downloaded and stored once, but also there would theoretically be no limit on how many instructions one flow entry can execute. By using the *goto-table* instruction within an instruction block, the table traversing order (i.e., the processing flow) can be dynamically changed. Instruction block update is also easy: one can simply load a new instruction block, update the block ID in affected flow entries, and then revoke the old instruction block if it is not needed anymore.

To facilitate instruction block sharing and at the same time enable differentiated flow treatment, we augment the flow entry with a parameter field. This field can be leveraged by application developers to define any parameters used by the associated instruction block. For example, in an egress table, when all the entries execute an output action, they may have different target output ports. While the output action is coded in an instruction block and shared by all the flow entries, the output port number is stored in the parameter field of each flow entry. This is just an overly simplified example. In reality, this mechanism is efficient in code space reduction.

We also abstract the globally-shared memory resource as a flow metadata pool. Flow metadata can be shared by flow entries to store statistics (i.e., counters) or any other information such as flow states. This is another enhancement on top of the existing packet metadata mechanism which is only dedicated to each packet. In particular, the expressivity of flow metadata enables stateful data-plane programming.

### B. Programming over OpenFlow Interface

Above the OpenFlow interface, any network forwarding application needs to be converted to the standard OpenFlow configuration commands and instruction blocks first. There are three ways to do it. First, it would be handy to use some high-level language to program network applications on devices. The high-level language provides another layer of abstraction that supports modularity and composition [10]. With the help of a high-level language, developers can focus on application functions rather than dealing with particular FE architecture and conducting error-prone table and flow manipulations. Some SDN programming languages have been proposed in literature [11], [12]. However, they are more focused on network-wide policy deployment on FEs built with fixed-function chips. Recently, a device-level programming language called P4 was proposed [13]. Some latest NPU chips are made C-programmable [14], [15], albeit only accessible by device developers. No matter which language is picked, a new compiler needs to be developed for sure. But, using a popular language can shorten the learning curve and increase the productivity. We are exploring the possibility of using C and Java as our choice of high level language. However, this is still an open and active research area. Until we thoroughly fathom the feasibility, we do not exclude other possibilities.

Although programming in a high-level language is meant to be forwarding-platform-independent, we realize that in the near future, many different forwarding architectures will coexist. For example, some chips (notably ASIC-based chips) have a physical front-end packet parser which parses packets in a centralized way but some other chips (notably NPU-based chips), for performance reason, prefer incremental packet parsing where packets are parsed layer by layer along the packet processing pipeline. Moreover, each kind of chip may have its own feature extensions, hardware-accelerated modules, and other nuances in hardware resource provisioning. Without discerning these differences, a generic program would pose significant challenges to the complier, which may lead to poor performance or even worse, failure to compile at all. Therefore, the application program should follow some programming style constraint upfront and may include some preprocessor directives to guide the compiling process. The key point is that the language itself must be general enough. The platform-independent compiler compiles the application programs by calling the platform-optimized library and generates an Intermediate Representative (IR), which will be passed down to FEs through the OpenFlow interface. This is not a perfect solution from a purist's perspective. However, as long as the FE chips do not converge to a single architecture, we have to live with it. The good thing is, if in the future the chip architectures do converge, the design flow and interface do not need to change.

Another method is to directly use Graphical User Interface (GUI)/Command Line Interface (CLI) for interactive and dynamic data plane programming at runtime. This could be considered similar to low-level programming in assembly language. Although it needs to handle flow level details, this method is fast and can fully explore the FE flexibility. The GUI/CLI can be used to handle fast runtime reconfigurations and can also be used to directly download compiled applications to FEs. We have implemented an open-source GUI to support this programming method [16].

At last, there are many prevailing network applications and forwarding processes today. For example, the basic Layer 2 (L2) switching and Layer 3 (L3) Internet Protocol (IP) forwarding are still widely used. It would be counterproductive to try to develop them again and again. Also, some applications on some particular target platforms may have been deeply optimized to achieve the best possible performance. It would be difficult for inexperienced developers to implement these applications with a similar performance. Therefore, precompiled applications can be provided in a library by any third party and directly used to program the network. Conceptually, this is in line with the TTP developed by Open Networking Foundation (ONF) Forwarding Abstraction Working Group (FAWG) [2]. Once the specifications of these library applications are standardized or publicized, any third party can develop and release them. Users can also maintain their private library and download the program through GUI or CLI.

Note that these programming approaches are not mutually exclusive. In other words, an application could be implemented through the simultaneous use of more than one approach. In a typical scenario, the basic forwarding process is either customized by using the high-level language or taken from a standard library application, and then GUI/CLI is used for library application download, dynamic runtime updates, and interactive monitoring.

### C. Programming Diversified Platforms

Once the program in the form of standard IR is conveyed to the FEs through OpenFlow messages, the FEs may have its own platform-dependent compiler which compiles or maps the program to its local structures. Note that this platform-dependent compiling process can also happen in controller. In this case, the burden on FEs is alleviated but the controller would need to retain extra knowledge about the target platform. The pros and cons of both options are still open to debate, but we believe our choice represents a clean architectural cut and is better for a coherent OpenFlow interface which can seamlessly support both configuration and operation. We roughly categorize FEs into four groups based on the type of main forwarding chips on them.

*1) Conventional ASIC-based:* Conventional ASICs for FEs typically have a fixed feature set and are not openly programmable. However, since they are designed to handle classical forwarding scenarios at high performance, they are still usable in SDN but in a more restrictive way. In this case, the standard library applications are the most suitable way to “program” the FEs. Some ASICs are configurable and can switch between different modes to support different applications. In this case, customized programming is not impossible but needs to be applied in a highly-disciplined way to ensure compatibility.

*2) SDN ASIC-based:* Recent research has started to pay more attention to SDN-optimized chips [3], [17]. Some companies are planing or have started to develop chips to better support flexible network application programming [4]. We can also put FPGA, if properly designed, into this category. These chips have embedded programmable capability for general packet handling but are also heavily populated with hardware-accelerated modules to handle common network functions for high performance. For these chips, it is feasible to use any kind of programming method. Due to the architectural limitations (e.g., hardware pipeline), low level interactive programming may not be well supported in these FEs. Therefore, the customized programming and application installation are preferred. A target-specific compiler is needed to compile the IR into the chip’s local structure.

The compiler, no matter how well-designed, may cause some performance loss due to the extra level of indirection. When the OpenFlow 2.0 is standardized, it is conceivable that in the future we could even design a chip that can natively execute the POF-FIS instructions without even needing a compiler in data plane.

*3) CPU-based:* CPU is no doubt the most flexible platform. Albeit having lower performance compared with the other platforms, it can easily support any programming method. Software-based virtual switches (e.g., OVS) are widely used in data centers. The switch implementation in CPU can basically run in two different modes: compiler mode and interpreter mode. The former compiles an application in IR into machine binary code (akin to the customized programming approach) and the latter requires the forwarding plane to directly interpret and execute OpenFlow instructions dynamically (akin to the interactive programming approach). The interpreter mode is more straightforward to implement and allows more flexible usage of the switch. The open source soft switch presented in [16] works in interpreter mode. It is unclear to us which mode has higher performance. We are working on a compiler-mode implementation based on x86 platform which targets on OVS.

*4) NPU-based:* NPUs are software programmable chips. They are designed specifically for network applications. An NPU typically contains multiple processing cores to enhance the parallel processing capability. NPUs can be broadly categorized into two types: pipeline and Run-To-Completion (RTC).

A representative pipeline NPU is EZchip’s NP family chip [18]. In a pipeline NPU, each stage processor only handles a portion of packet processing tasks. Although the pipeline NPU’s architecture seems to match OpenFlow’s processing pipeline model, in reality it is not easy to perfectly map the two pipelines together because OpenFlow’s pipeline is function-oriented and NPU’s pipeline is performance-oriented. The compiler needs to carefully craft the job partition to balance the load of pipeline stages.

In an RTC NPU, each processor core is responsible for the entire processing of a packet. This architecture maximizes the programming flexibility, which is similar to CPUs. However, it has limited code space per core and needs to share resources (e.g., memory) among cores. The code space constraint requires the code size to be compact enough in order to accommodate the whole processing procedure (e.g., we cannot afford to repeat the storage of the same set of actions for every flow in a large flow table). The resource sharing constraint requires both the number of memory accesses and the transaction size per memory access to be minimized in order to meet the performance target. Fortunately, the new features we proposed for the OpenFlow 2.0 interface allow software developers to program efficiently with these constraints in mind.

NPU-based FEs can also be programmed in compiler mode or interpreter mode. In the next section, we discuss the implementations of both modes on an NPU-based FE and compare their performance.

## III. NPU-based Case Study

The NPU-based FE prototype works on Huawei’s NE-5000 core router platform. The line card we used has an in-house designed 40G NPU and each half slot interface card has eight 1GbE optical interfaces. The multi-core NPU runs in RTC mode.

### A. Forwarding Programming in C

To support high level data plane programming, we model three entities: Metadata, Table, and Packet. The program simply manipulates these three entities and forwards the resulting packets based on the table lookup results. For our NPU, the three entities are all realized in registers. Metadata is used to hold the packet metadata which is represented as a customized structure; Table is the associated data of flow entries loaded from table matches, which is also represented as a customized structure; Packet is typically the packet header under process which is described in another structure.

Figure 3 shows the structures of Metadata, Table, and Packet for an L3 forwarding application. A piece of program that processes a packet is shown in Figure 4. It combines the IP address and the Virtual Private Network (VPN) ID as a new key to conducts another table lookup.

Once the packet processing flow is described in C, it is straightforward to compile the program into IR, which include protocol parsing rules, table specifications, and flow actions.

```
struct Metadata_L3 {
    uint8 L3Stake; //L3 Offset
    uint16 VpnID; //VPN ID
    uint16 RealLength; //Packet Length
    uint16 SqID; //QOS Queue ID
};
struct Table_Portinfo {
    uint16 VpnID; //VPN ID
    uint16 SqID; //QOS Queue ID
};
struct IPV4_HEADER_S {
    uint4 Version;
    uint4 HeaderLength;
    union {
        uint8   TOS;
        uint6   DSCP;
        uint3   Precedence;
    };
    uint16 TotalLength;
    uint16 FragReAssemID;
    IPV4_FRAG_HWORD_S FragHWord;
    IPV4_TTL_PROT_HWORD_S TtlProtWord;
    uint16  Checksum;
    uint32  SIP;
    uint32  DIP;
};
```

Figure 3: Structures for L3 Forwarding

```
(Metadata_L3 *) p_metadata;
(Table_Portinfo *) p_table;
p_metatada->VpnID = p_table->VpnID;
p_ipheader = p_packet + 14;
Goto_Table(TableID, p_metadata->VpnID, p_ipheader->DIP);
```

Figure 4: Code Example

Although the programming style appears to be platform independent, the Goto_Table library function could be specific for each different forwarding platform in the above example. To infer the different platform implementation to the compiler, an NPU-specific proprietary library is included.

### B. Interpreter Mode FE Implementation

In interpreter mode, each POF-FIS instruction in a flow action (i.e., an instruction block) corresponds to a piece of code written in NPU microcode which realizes the instruction's function. The code translation is straightforward. However, due to the flexibility embedded in the POF-FIS instructions, the efficiency of the microcode is problematic.

For example, the Goto_Table instruction may lead to a complex microcode processing flow. First, it needs to read the corresponding table information and initialize a buffer to hold the search key, then it enters a loop to construct the search key piece by piece depending on the number of header fields involved in the instruction. Each iteration of the loop contains many steps. It needs to locate the target field using the offset and length information, copy the field into the key buffer, and mask the field. This process requires a lot of pointer shift, data move, and other logic operations. Finally, the search key is sent to the target flow table and the thread is hung up to wait for the lookup result.

The inefficiency comes from three sources: (a) the microcode instruction count, (b) the number of thread switch, and (c) the bandwidth of loading flow table entries. The microcode instruction count is determined by the microcode instruction set and the complexity of POF-FIS instructions. The thread switch is caused by the loops that force to break processing pipelines, as well as the latency for table lookups. Each table lookup will return an instruction block. If parameters are directly carried within instructions, the bandwidth of loading such instruction blocks are considerably expanded. As a result, the throughput suffers. The last inefficiency can be addressed by allowing the flow entry to carry the parameters but this is not enabled in our prototype yet.

In general, the interpreter mode implementation is suitable for the interactive programming approach in which the data path is fluid and can be constantly changed. While this mode is less likely to be widely used in production networks, it is interesting in experimental and research environments for quick design verification.

### C. Compiler Mode FE Implementation

In compiler mode, the application is considered a whole and a relatively static entity. This allows the compiling process to simplify the microcode. Since there are a set of registers $R0 \sim Rn$ in NPU, the compiler can resolve the pointer offsets and directly map the data into registers. This eliminates the need of pointer manipulations in microcode. The compiler also handles the length evaluation and directly translates that into assignment statement. These can help to reduce the microcode instruction count by more than 50%.

The compiler mode implementation can easily take advantages of the flow parameter mechanism which reduces the instruction block size. This lowers the bandwidth requirement for memory access and further boosts the throughput and latency performance.

### D. Performance Evaluation

The packet forwarding performance in NPU is evaluated by throughput ($R$) and packet latency ($L$). We know that $R = c * f/i$ and $L = t/R$ in which $c$ is the number of processing cores, $f$ is core frequency, $i$ is microcode instruction count per core, and $t$ is the number of threads. Given an NPU, $c$ and $f$ are fixed, so the performance is mainly determined by $i$ and $t$. Reducing table lookup latency and memory access bandwidth have direct impact on $t$. Table I compares the performance of different Goto_Table implementations ($n$ is the number of match fields in the search key).

TABLE I: *Goto_Table* Performance Comparison

| | instr. count | # thread switch |
|---|---|---|
| Interpreter Mode | $37 + 33n$ | $7 + 3n$ |
| Compiler Mode | 13+n | 1 |

Table II summarizes the performance comparison for basic IPv4 forwarding. The conventional non-SDN implementation is used as a benchmark, which has exactly the same function as the SDN-based implementations. The conventional implementation can fully exploit the hardware features (e.g., protocol parsers) and the microcode is deeply optimized.

Through extensive experiments, we found that the compiler-mode implementation performs consistently better than the interpreter-mode implementation. For a typical IP forwarding process in routers, the compiler-mode implementation

TABLE II: Performance Comparison for *IPv4* Forwarding

| | non-SDN | Interpreter | Compiler |
|---|---|---|---|
| instr. count | 496 | 1089 | 550 |
| # thread switch | 94 | 146 | 74 |
| thruput (Mpps) | 77.5 | 35.3 | 69.8 |
| latency (cycle) | 4468 | 6361 | 4022 |

needs 57% less microcode instructions than the interpreter-mode implementation. Compared with the conventional implementation, the compiler-mode implementation is just 11% worse. With the same number of micro cores, a compiler-mode implementation can easily double the throughput of an interpreter-mode implementation.

## IV. Related Work

This paper is concerned with the SDN data plane programming issues. To put it in context, interested readers can refer to Kreutz et al. [5] for an up-to-date comprehensive survey of SDN research and practice.

*P4* language is based on an abstract forwarding model [13]. The use of *P4* is akin to our customized programming approach. For an application, it defines the header parse graph and the switch control program. The control program basically describes the tables, the action set supported by each table, and the table dependencies. The model also needs a platform-dependent compiler to map the configuration to each specific target switch. After configuration, the controller can then populate the tables with actual flow entries at run time.

Open Compute Project (OCP) networking project advocates open switches with open-programming environments [19]. Quite a few open switch specifications and open-source softwares have been released since the project debut in 2013. However, at its current stage this project still falls short of SDN support: (1) It focuses on programming in an open Linux-based Network Operating System (NOS) environment for each individual switch but not in a centralized SDN programming environment; (2) The current open switch specifications heavily rely on existing ASIC-based chips and Software Development Kit (SDK)/Application Programming Interface (API) provided by chip vendors. The programming flexibility is limited by the chip architecture and the degree of openness the chip vendors would like to offer. We believe a truly open switch also means open silicon chips or at least a universal and complete API. The project might evolve towards a similar direction as we proposed.

## V. Conclusion and Future Work

We believe it is plausible to assume that the next generation SDN will require total programmability over an open data plane. An FE could be programmed as easily as a bare-metal server can be programmed today. However, the diversified chips used to build the FEs today and in the foreseeable future are far from a convergence. This poses a challenge for the desired uniform and coherent SDN programming experience. Until we solve this problem, we cannot claim a vertical-decoupling of the SDN layered architecture is fully achieved. With the current SDN approach, it could become difficult to build an efficient ecosystem in which players would work at different layers independently.

In this paper, we presented our initial exploration and experience on this hard problem. We propose a programming framework which centers on the next-generation OpenFlow interface, targets various FEs, and supports different programming approaches. In particular, we experiment on an NPU-based platform and show that the complier-mode implementation is superior to the interpreter-mode implementation in terms of performance, although interpreter mode implementation offers much better runtime flexibility.

Our future work includes completing the proposed SDN programming framework by implementing the missing pieces in Figure 1 (e.g., platform-dependent compilers for other FE platforms) and demonstrating real-world SDN applications through the full programming process. We are also working on extending OVS to support POF and making it runnable in Mininet environment, so the idea is more accessible to the research community. This programming framework can be considered as a proposal for the OpenFlow 2.0 standard.

## References

[1] Open vSwitch, http://openvswitch.org/ [retrieved: March, 2015] .

[2] ONF Forwarding Abstraction Working Group (FAWG), https://www.opennetworking.org/working-groups/forwarding-abstractions [retrieved: March, 2015].

[3] Pat Bosshart et al., “Forwarding Metamorphosis: Fast Programmable Match-action Processing in Hardware for SDN,” in *Proceedings of the ACM SIGCOMM*, 2013, pp. 99-110.

[4] White Box Week: Chip Startups Take Aim at Broadcom, https://www.sdncentral.com/news/white-box-week-chip-startups-take-aim-broadcom/2013/11/ [retrieved: March, 2015] .

[5] Diego Kreutz and Fernando Ramos and Paulo Esteves Verissimo and Christian Esteve Rothenberg and Siamak Azodolmolky and Steve Uhlig, “Software-Defined Networking: A Comprehensive Survey,” *Proceedings of the IEEE*, January 2015, pp 14-76.

[6] Software Defined Specification Environment for Networking, http://www.xilinx.com/applications/wired-communications/sdnet.html [retrieved: March, 2015] .

[7] Nick McKeown et al., “OpenFlow: Enabling Innovation in Campus Networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, April 2008, pp. 69-74.

[8] Haoyu Song, “Protocol-Oblivious Forwarding: Unleash the Power of SDN through a Future-Proof Forwarding Plane,” in *ACM SIGCOMM HotSDN Workshop*, 2013, pp. 127-132.

[9] Jingzhou Yu and Xiaozhong Wang and Jian Song and Yuanming Zheng and Haoyu Song, “Forwarding Programming in Protocol-Oblivious Instruction Set,” in *IEEE ICNP CoolSDN Worshop*, 2014, pp. 577-582.

[10] Nate Foster et al., “Languages for Software Defined Networks,” *IEEE Communication Magazine*, Feburary 2013, pp. 128-134.

[11] ——, “Frenetic: A Network Programming Language,” in *ACM SIGPLAN ICFP*, 2011, pp. 279-291.

[12] Andreas Voellmy and Paul Hudak, “Nettle: Taking the Sting Out of Programming Network Routers,” in *PADL*, 2011, pp. 235-249.

[13] Pat Bosshart et al., “P4: Programming Protocol Independent Packet Processors,” *Computer Communication Review*, 2014, pp. 87-95.

[14] EZchip NPS, http://www.ezchip.com/ [retrieved: March, 2015].

[15] Netronome Flow Processor, http://www.netronome.com/ [retrieved: March, 2015].

[16] Protocol Oblivious Forwarding, http://www.poforwarding.org [retrieved: March, 2015].

[17] Martin Casado and Teemu Koponen and Daekyeong Moon and Scott Shenker, “Rethinking Packet Forwarding Hardware,” in *ACM SIGCOMM HotNets Workshop*, November 2008, pp. 1-6.

[18] Ran Giladi, “Network Processors: Architecture, Programming, and Implementation (Systems on Silicon),” *Morgan Kaufmann*, 2008.

[19] Open Compute Project, http://www.opencompute.org/ [retrieved: March, 2015] .

# Compass: A Data Center Bootloader for Software Defined Infrastructure

Shuo Yang, Weidong Shao, Haoyu Song

Huawei Technologies
Santa Clara, CA, USA, 95124
email: {shuo.yang, weidong.shao, haoyu.song}@huawei.com

Wei Xu

Tsinghua University
Beijing, China, 100084
email: weixu@mail.tsinghua.edu.cn

***Abstract*—In this paper, we present a design of data center deployment automation system, *Compass*, for bootstrapping a software defined infrastructure, including network and compute nodes. *Compass* automates the process of bootstrapping a Software-Defined Networking (SDN) -based network from bare-metal networking devices and provisioning the bare-metal servers through the SDN network in a unified management approach. The unified and streamlined deployment management of networking and compute resources not only reduces the initial deployment cost, but also provides a way to automatically scale out data center infrastructure's capacity horizontally after the initial infrastructure setup (e.g., adding networking and computing resources, etc.). Using *Compass*, we have deployed a private cloud in a medium scale data center with around 200 commodity servers and over 20 SDN switches in Tsinghua University. We present the case study in this paper to illustrate the benefits of Compass as a unified deployment and management tool in a data center's software define infrastructure.**

***Keywords–Compass; data center; bootloader; automation; SDN.***

## I. INTRODUCTION

Servers and network devices (e.g., switches and routers) are key components of the data center infrastructure. In the past, the two parts are usually provisioned and managed by different teams using different tool sets. This situation poses several challenges in efficient system planning, optimization, and debugging, and in turn incurs high operational cost and low return on investment.

Ideally, the two parts should be treated as an integral entity. But in reality, they are handled separately from administrative perspective. There are also some technical reasons for this separation. One reason is that networking devices are vertically integrated, and the deployment procedure is handled very differently from server management. However, we observed several appealing trends in industry which can change the current status quo.

First, there is a trend of "software defined everything" movement. Server software has a long history of being 'software defined'. Software-Defined Networking (SDN) [1] has prevailed not only in academic research but also in industry adoption. Moreover, storage industry starts the 'software defined' roadmap and practice [2], [3]. IBM coined the term "software defined environment" to address the vision for automatic and dynamic computing infrastructure provision [4]. The trend becomes increasingly clear as the concepts and practices such as Software-Defined Data Center (SDDC) [5] and warehouse-scale computing [6] prevail. The entire data center can be modeled as one big computer comprised of distributed computing/storage nodes, which in turn are interconnected through a network fabric comprised of switches and routers.

Another trend we see is the "open everything" movement. Not only the entire software stack for applications [7], [8], operating systems [9], and cloud managements [10] have been opened up, but also the hardware itself [11]. On the one hand, the data center building blocks are all standard-based. The choices of both software and hardware to construct the data center are abundant and cost efficient. On the other hand, these choices can become overwhelming for data center operators. The sheer scale of the open ecosystem can be daunting even to experienced Information Technology (IT) staff. There is apparently a lack of a capable orchestrator which can glue every piece of the system together organically and make them run in concert.

The interesting question we would like to answer is: Can we consolidate the best tools and automate the deployment of entire data center infrastructure in a seamless way? To be specific, we assume a greenfield deployment of new data center with pure bare-mental servers and switches. The only need from IT staff is to physically wire the devices together according some topology plan and then power up the data center. What if the data center administrators manage the modern software defined infrastructure deployment in the same way as a Linux system admin does today with a bootloader?

Though these questions sound like system administration related, we argue that they are critical if the industry wants to adopt new SDN technologies. As everything is software defined and both software and hardware are open, a 'bootloader' at data center scale is needed to deploy the software efficiently and coherently to various commodity hardware resource. A solution toward this will enable a unified portal and a coherent method to configure and manage the entire data center infrastructure, just as we do today for installing software components and services features onto a computer.

In this paper, we present the scheme and open source tool we developed, *Compass*, to enable a unified software defined infrastructure. We also share our successful experience on actual deployment of a data center in Tsinghua University, which has around 200 servers and over 20 SDN switches in the first phase. Our experience is more from 'out-of-box' system administration's perspective in greenfield SDN adoption scenario. We demonstrate that even a small step toward data center bootloader can significantly reduce the roadblock of SDN adoption in the context of entirely software defined infrastructure in data center, and we demonstrate that IT administration needs a 'think-out-of-box' methodology in this 'software defined everything' and 'open everything' era.

What we add is a unified bootloading system that works

ISBN: 978-1-61208-404-6

on both switches and servers. After proper operating systems are installed and booted up, the switches and servers are further configured with the state-of-art open-source software, which can monitor and control the computing, storage, and networking components of the data center infrastructure.

The benefit of our approach is tremendous. It can significantly reduce the data center operation and maintenance cost. It allows the data center operators to focus on their core business, that is, to provide better data services to customers. It enables zero-touch scaling of the existing data center. New software and new hardware can be incrementally deployed without disturbing the normal data center operation.

This paper makes the following contributions:

First, it presents a unified deployment management system design, which will consolidate the infrastructure bootstrapping process. Traditionally, network infrastructure deployment and server/storage infrastructure deployment are considered separate efforts, which normally results in separate teams and prolonged engineering schedule. We present the first deployment system that unifies the above procedures and fills the gap in between in the era of SDN.

Second, it describes in details the first open system, *Compass*, which reflects the above vision. The novelty of *Compass* can be summarized as follows. (a) *Compass* is an open system not only in the sense that it is open sourced [12], but also in the sense that it is open to existing building blocks such as configuration management tools and OS provisioning tools through pluggable interfaces. (b) We present our engineering experience of quickly bootstrapping an SDN infrastructure and private cloud with a unified viewpoint.

The remaining of the paper is organized as follows. Section II describes the high level architecture of the data center and the procedure to deploy it. Section III explains the building blocks of our tool and the benefits of our design in greater details. Section IV presents the real word deployment of a data center and share our preliminary performance evaluation results. Section V compares our work with some existing related work, and finally, Section VI concludes the paper and suggests the future work.

## II. High Level Architecture

In this section, we describe the high level architecture of an SDN-enabled data center and the role *Compass* plays in this architecture. We partition the data center into three tiers: controller tier, network tier, and server tier, as shown in Figure 1. The controller tier hosts all the tools that are required to configure and control the data center as well as all the software that will be installed on the switches and servers. Note that the controller tier contains servers for three different roles. These roles are relatively independant and can be realized on the same or different physical machines. Before actually deploy the data center, a data center blueprint should be prepared to specify the network address scheme, target service locations, virtualization scheme, network topology, and bandwidth allocations.

The network tier includes all the switches, which can be roughly mapped to the Fabric Elements described by Casado et al. [13]. The switches can be arranged in any topologies such as Clos, Fat Tree, and 3D Torus. In our design, we use the fat tree topology and a software-defined networking architecture for which the switch behavior is programmed and controlled by an OpenFlow [14] controller.

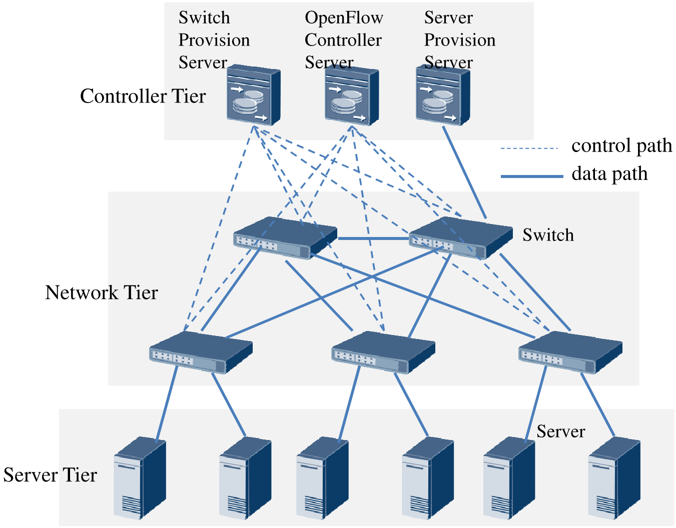


Figure 1: Architecture of Software Defined Data Center Infrastructure

The server tier includes all the servers. There are two types of configurations at this tier. One is to use virtual switches, such as Open Virtual Switch (OVS) [15], on the servers to enable edge intelligence. In this case, *Compass* will deploy a virtual switch on each server and configure the control plane connectivity to the same OpenFlow controller, which controls the overall network fabric. This step essentially extends the network tier into the server tier through virtualization. In this case, there should also be dashed lines from the OpenFlow controller server to all the servers in Figure 1. In the other type of configuration, virtual switches are not used. A server could still spawn multiple virtual machines, but all the networking packets to and from these virtual machines will be switched by the physical switches in the network tier.

The workflow is as follows. When the data center is powered up, the network switches are first installed with a Network Operating System (NOS) with OpenFlow agent enabled. This is done through the switch's control interface. After the NOS boots up, the OpenFlow agent hands over the switch control to OpenFlow controller. The controller first learns the network topology and then starts to configure the network. The controller conducts the network sanity check, partitions the virtual networks, configures the gateways, and provisions the flows and flow bandwidths. Once the network is configured, all the servers are reachable. The server Operating System (OS) and application software are then installed over the network. If virtual switches are installed in this step, then the OpenFlow controller must first configure these virtual switches to make the virtual machine reachable. Each virtual machine can then be provisioned individually.

## III. System Design of Compass

To fulfill the vision and showcase a workable system described in previous sections, we designed *Compass*, a data center bootloader. It provides a unified view and workflow for the networking and server infrastructure deployment process of a software defined data center. Programmability and extensibility are the primary goals our design aims to achieve.

As shown in Figure 2, *Compass* provides six core com-

ISBN: 978-1-61208-404-6

ponents: RESTful Application Programming Interface (API) engine, Resource Discovery engine, OS Provisioning engine, Package Deployment engine, Messaging engine, and Data Persistence engine. Here is how they work together as a system. Through the Restful API engine, *Compass* user can specify how they would like to design the software defined data centers, i.e., what the result system looks like. Resource Discovery engine provides the functionality to automatically discover hardware resources with corresponding network topology information in the data center once they are physically rack-and-stacked. For example, *Compass* uses SNMP protocol to query MIB table on switches to figure out the server MAC addresses connected to specific networking devices. Since each machine will send ARP requests to a switch during bootstrapping process, this approach gives *Compass* a view of all computing devices. Moreover, as long as the computing resources are rack-and-stacked following well-defined rules, *Compass* can translate its port position information into the location information; therefore a physical to logical mapping is created. OS Provisioning engine can install the specified operating system or hypervisor accordingly, if needed, onto those physical resources. Package Deployment engine will specify the service package for different building block and properly configure them into the functioning states. The above engines communicate with each other through the Messaging engine, so that everything is push-based event driven. This design makes *Compass* orchestrate the complex deployment process of whole data center like a symphony. Last but not least, the Data Persistence engine is used to store the states for *Compass* to act properly at each step.

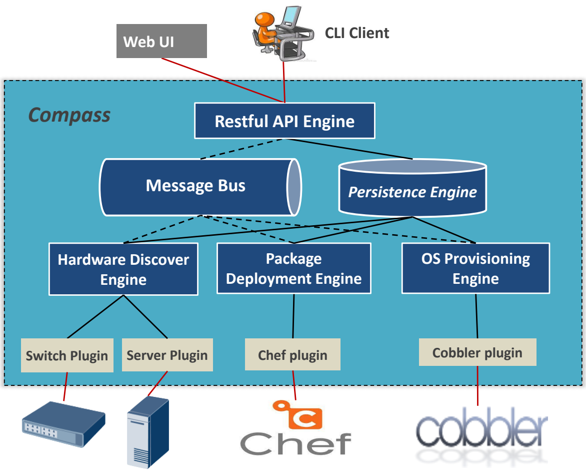


Figure 2: Compass Software Function Compoments

### A. Programmability

*Compass*'s programmability is enabled through a RESTful API engine. A set of APIs provide a programming contract to external applications. Each Uniform Resource Identifier (URI) [16] in our RESTful API corresponds to a resource object in data center infrastructure. A resource object supports operations such as create, update, delete, and execute actions. In addition, *Compass* provides a client-side Python library as a RESTful API wrapper to facilitate complex data center deployment scenario. A user can configure and deploy a cluster through Command Line Interface (CLI) scripts.

Table I shows a subset of resource objects. */switches* and */machines* represent the hardware resources *Compass* discovers and deploys the target systems to (such as Haddop or OpenStack). */drivers* is a metaphor we borrowed from OS community, and it defines the final software infrastructure that *Compass* bring the data center into. */drivers* is composable in a data center, i.e., *Compass* can bring a data center into an OpenStack [10] compute cluster along with Ceph [2] storage cluster and with Pica8 [17] SDN switches as the networking fabric. */clusters* and */clusterhosts* are the resource objects describing the resulting systems. Note that we only show a subset of resource objects here to illustrate the programming capability of *Compass*. The other resource objects are omitted here for conciseness. Interested reader can refer to the *Compass* website [18] for detailed documents.

TABLE I: Example URI of Compass API

| URI | Resource | Operations |
|---|---|---|
| /switches | Networking switches | create, edit |
| /machines | Physical servers | create, edit |
| /drivers | An installer of a target system(e.g, OpenStack) | create, list |
| /clusters | A cluster with a target system to be installed | create, edit, delete |
| /clusterhosts | A host in a cluster | create, edit, delete |

### B. Extensibility

The Resource Discovery engine, the OS Provisioning engine, and the Package Deployment engine are the heavy-lifting internals of *Compass*. The Resource Discovery engine can collect the physical resources that are connected to a management plane and understand the network topology through protocols, such as Link Layer Discovery Protocol (LLDP) and the resource capability through mechanisms such as Ohai [19]. The Resource Discovery engine updates the hardware cluster status in the Persistence engine and notify the other components through the Messaging engine.

After the hardware discovery is done, the OS Provisioning engine is able to deploy the corresponding operating system or hypervisor to the physical nodes following the setup instruction stored in the Persistence engine. Note that OS or hypervisor setup instruction is defined by *Compass* user through the RESTful API engine. Therefore, the behavior for this step is programmable. The current *Compass* implementation uses Cobbler [20] as the actual OS provisioning tool. Cobbler is integrated into *Compass* as an OS Provisioning engine driver plug-in. And then the Package Deployment engine follows the setup instruction or policy-based rules stored in the Persistence engine to deploy software components onto the resource nodes. Moreover, the Package Deployment engine is in charge of the proper configuration, such as setting up SDN controller IP and trust credentials on the SDN switches, so that the deployed distributed systems function as a logical cluster as they are designed. The current *Compass* implementation uses Chef [21] as the actual configuration management tool, and Chef is integrated as a package deployment driver plug-in.

As we can see from the above, *Compass* takes a 'microkernel' software architecture and uses plug-in mechanism to delegate works to the actual 'drivers'. In this way, it can provide extensibility in the following dimensions with *minimal plug-in development*:

- It is extensible with regard to the server hardware that it can support, be it Dell, HP, or Open Compute Project (OCP) [11] servers.
- It is extensible with regard to the switch hardware that it can support, be it Pica8, BigSwitch, or OCP [11] switches.
- It is extensible with regard to the target systems that it can configure, be it an SDN enabled OpenStack cloud or SDN enabled distributed file system cluster such as Ceph and Hadoop cluster.
- It is extensible with regard to the network OS it can provision, be it PicaOS [17], BigSwitch Open Network Linux (ONL) OS [22], or Cumulus OS [23]; and it is extensible with regard to the server OS or hypervisor it can provision, be it CentOS, Ubuntu, or even ESXi.

Because of the above design principle, *Compass* code base is ‘small’. It is around 6000 line of Python code at its core and the complexity of extending plug-in is low. Here is an example of extensibility through *minimal plug-in development* effort at the dimension of resource discovery. We supported Huawei switch for server hardware auto discovery in our initial development. During our real deployment scenarios, we encountered Pica8 switches and Arista switches. Because of our ‘microkernel’ architecture and plug-in interface, we were able to add just about 200 line of Python plug-in code to support these switches and achieve the same functionality.

## IV. Experiment and Evaluation

In this section, we share our experience using *Compass* to deploy a private OpenStack [10] cloud with Pica8 [17] SDN switches as the network tier in Tsinghua University. We used the OpenStack’s Grizzly release with Neutron as the cloud virtual networking provisioning engine.

OpenStack is an exemplification of the level of complexity of the modern software defined infrastructure. It requires not only configuring the networking infrastructure but also configuring the server infrastructure. Configuring a production OpenStack cloud is notoriously deemed as a maze for most system administrators. Moreover, the majority of OpenStack issues originated from the networking misconfiguration. The number of Neutron (networking) related configuration is around 100, and the number of Nova (compute) related configuration is around 150. It is extremely hard if not impossible for administrators to properly configure the system in a productive way.

Figure 3 shows how *Compass* deploys the entire cloud system not only the SDN networking infrastructure, i.e., the network tier described in Figure 1, but also the distributed system on the server tier. In the deployment process, the operator uses the web User Interface (UI) we provide to follow the step-by-step configuration – these steps essentially reflect the operator’s thought process toward the design of the whole software defined data center. The web UI talks to our RESTful API engine to persist the design decisions for the heavy lifting components to make decision for the real deployment command and control process. It first deploy the Pica8 switches, which is the network tier as described in Figure 1. Currently, PicaOS does not allow OS provisioning. Therefore our OS provisioning step is skipped (as a no-op) at this moment. The *Compass* Package Management engine kicks in right after the switches are discovered. During the package management process, *Compass* orchestrates Pica8 switches to configure themselves into the proper service state and establish connectivity to their controllers. We are working with Pica8 for OS provisioning using the Open Network Install Environment (ONIE) [24] approach, so that other OS, such as Debian, can be provisioned to the bare-metal switches. In the long run, we envision ONIE will be adopted by ‘open switches’, and therefore, the Compass logic can bootstrap switches from bare-metal. After this step, *Compass* proceeds to work on the servers including OS provisioning and package deployment of server software such as OVS and OpenStack management agents.

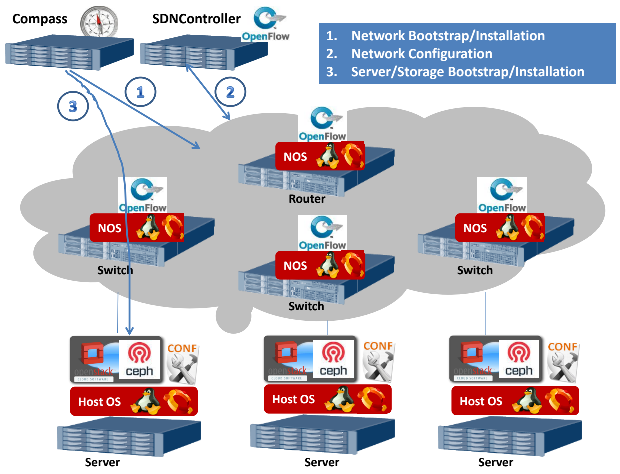


Figure 3: A Unified Process of the Entire Software Defined Infrastructure Deployment

In our real deployment procedure, *Compass* helped the operator find the way out of the above maze. Instead of configuring totally 250 parameters, the operator only needs to program the RESTful API server through the web UI with 6 wizard-based steps. Our deployment process took a little over one hour to deploy the entire SDN enabled OpenStack cluster, with over 20 Pica8 switches and around 200 physical rack servers, from bare-metal to a fully functional OpenStack cloud.

## V. Related Work

ONIE [24] is an open source project which solves the networking device OS installation problem. Here are some issues of ONIE. First, ONIE only works for network switch deployment and it does not provide a unified mechanism for both switch and server deployment. Secondly, ONIE provides the automation at individual device level, i.e., to benefit from ONIE, the bare-metal switch is required to pre-install a special boot-loader image while our tool eliminates this requirement (our current practice is taking a non-ONIE in the loop approach). ONIE naturally fits into *Compass* extensible plug-in architecture, in which *Compass* can leverage its capability to bring a global view of entire networking infrastructure deployment. We are working on the ONIE plug-in extension for ONIE enabled networking devices.

Some long standing software deployment solutions, such as Rembo Preboot eXecution Environment (PXE) [25], IBM Tivoli [26], and Symantec Altiris [27], include the bootstrap of operating systems on a diversity of hardware architectures. However, these solutions all assume the networking infrastructure is ready for server software deployment.

Crowbar [28], an deployment automation project lead by Dell, assumes that network infrastructure has been deployed before it takes over the server software deployment process. This is a reasonable assumption in traditional data centers where vertically integrated networking boxes were the only option for networking infrastructure – an old paradigm that data center builders did not have an option to deploy software defined networking infrastructure. But as we have described, these assumptions do not hold any more. *Compass* is different from Crowbar as it designed for not only server infrastructure deployment but also for networking infrastructure deployment through a unified viewpoint.

Fuel [29] and TripleO [30] are tools tightly designed for deploying OpenStack cloud management platform, while *Compass* is designed for extensible capability toward software defined infrastructure deployment. We extended *Compass*'s capability to support Ceph deployment through a Ceph driver while keeping the rest of code unchanged (see III-A for the driver concept). Configuration management tools such as Chef [21], Puppet [31], and Ansible [32] provide configuration capability for software. However, they do not provide resource provisioning capability, which is a key step in data center deployment. As we described, *Compass* is open to utilized these tools as components of its automation process and it delegates the configuration management functionality to these existing tools to avoid re-inventing wheels. Specifically, our current implementation use Chef as our configuration management plug-in and we are working toward an Ansible plug-in.

## VI. Conclusions

The hardware and software decoupling is becoming the new norm of the network camp thanks to the advent of the software defined networking and the open network movement. The ubiquitously available open-source software and bare-mental devices gives data centers unforeseen opportunity to optimize their cost structure and provide agile services at scale. Finally, we are able to program and control network devices just like we program and control a server. Then, why would we still need to use two different tool chains and skill sets to deploy and manage servers and networks?

In this paper, we revisit the data center provision problem. We treat the entire data center infrastructure as an organic entity and use a unified tool to deploy and provision the data center from a scratch. SDN unlocks network management in a great simplicity, and our system utilizes that promise and demonstrates the real benefit in a greenfield case study from system administration perspective. Our approach is fundamental in that it consolidates the two historically separate worlds together and significantly simplifies the data center infrastructure deployment. As far as we know, this is the first such tool available with this vision in industry.

*Compass* is just a start of the effort of building a unified management tool for software defined infrastructure. Several works are ongoing. As *Compass* is a 'microkernel' design and can be easily extended in functionality, we are working on ONIE integration for switches pre-installed ONIE when they are shipped to data center. We are working on handling the server like PXE booting sequence for networking devices as another *Compass* plug-in. All of these promise the unification mechanism for both server and networking devices.

Thanks to the SDN evolvement, the last locked infrastructure in data center is now opening up. Data center infrastructure deployment is just the start. Our tool can be extended and integrated with other tools to continue managing, monitoring, and controlling the data center. For example, servers and switches can be upgraded or replaced without interrupting the data center services. Our tool should be able to schedule and automate the tasks with minimum human interference. While we leave these as our future work, we are confident that the data center automation would become the first-class requirement in building, running, and maintaining a data center. It is our hope to evolve *Compass* to become an invaluable tool for future cloud providers.

## VII. Acknowledgment

We would like to thank Sam Su, Xiaodong Wang, Xicheng Chang, Grace Yu, Jiahua Yan for participating the design and implementation of *Compass*.

## References

[1] D. Kreutz et al., "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, January 2015, pp 14-76.

[2] S. A. Weil, S. A. Brandt, E. L. Miller, D. E. Long, and C. Maltzahn, "Ceph: A Scalable, High-Performance Distributed File System," in *OSDI*, 2006, pp. 307-320.

[3] E. Thereska et al., "IOFlow: A Software-defined Storage Architecture," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, 2013, pp. 182-196.

[4] Software Defined Environment. [Online]. Available: http://www.ibm.com/systems/infrastructure/us/en/software-defined-environment/ [retrieved: March, 2015]

[5] The Journey Toward the Software Defined Data Center. [Online]. Available: http://www.cognizant.com/InsightsWhitepapers/The-Journey-Toward-the-Software-Defined-Data-Center.pdf [retrieved: March, 2015]

[6] L. A. Barroso, J. Clidaras, and U. Holzle, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 2nd ed., ser. Synthesis Lectures on Computer Architecture. Morgan Claypool Publishers, 2013.

[7] K. Douglas and S. Douglas, *PostgreSQL*. Thousand Oaks, CA, USA: New Riders Publishing, 2003.

[8] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[9] A. Kivity, "KVM: the Linux Virtual Machine Monitor," in *OLS '07: The 2007 Ottawa Linux Symposium*, July 2007, pp. 225-230.

[10] OpenStack: The Open Source Cloud Operating System. [Online]. Available: http://www.openstack.org/software/ [retrieved: March, 2015]

[11] Open Compute Project Community. [Online]. Available: http://www.opencompute.org/ [retrieved: March, 2015]

[12] Compass Github Entry. [Online]. Available: https://github.com/stackforge/compass-core [retrieved: March, 2015]

[13] M. Casado, T. Koponen, S. Shenker, and A. Tootoonchian, "Fabric: A Retrospective on Evolving SDN," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN)*, 2012, pp. 85-90.

[14] N. McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, April 2008, pp. 69-74.

[15] B. Pfaff et al., "Extending Networking into the Virtualization Layer," in *Proc. of workshop on Hot Topics in Networks (HotNets-VIII)*, 2009, pp. 1-6.

[16] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, 2000, aAI9980887.

[17] Pica8 Switch and Network OS. [Online]. Available: http://www.pica8.com/ [retrieved: March, 2015]

[18] Automating Distributed Systems Deployment. [Online]. Available: http://www.syscompass.org/ [retrieved: March, 2015]

[19] Document of Ohai. [Online]. Available: http://docs.opscode.com/ohai.html [retrieved: March, 2015]

[20] Cobbler Project Website. [Online]. Available: http://www.cobblerd.org/ [retrieved: March, 2015]

[21] Online Documentation for Chef. [Online]. Available: https://wiki.opscode.com/ [retrieved: March, 2015]

[22] BigSwitch Controller and Network OS. [Online]. Available: http://bigswitch.com/ [retrieved: March, 2015]

[23] Cumulus Linux. [Online]. Available: http://cumulusnetworks.com/product/overview/ [retrieved: March, 2015]

[24] Open Network Install Environment. [Online]. Available: http://onie.github.io/onie/ [retrieved: March, 2015]

[25] REMBO: A Complete Pre-OS Remote Management Solution REMBO Toolkit 2.0 Manual. [Online]. Available: http://goo.gl/DRQOdl [retrieved: March, 2015]

[26] IBM Tivoli. [Online]. Available: https://www.ibm.com/software/tivoli [retrieved: March, 2015]

[27] Endpoint Management powered by Altiris Technology. [Online]. Available: http://www.symantec.com/endpoint-management/ [retrieved: March, 2015]

[28] The Crowbar Project. [Online]. Available: http://crowbar.github.io/home.html [retrieved: March, 2015]

[29] Mirantis Fuel Project. [Online]. Available: http://software.mirantis.com/key-related-openstack-projects/project-fuel/ [retrieved: March, 2015]

[30] OpenStack on OpenStack. [Online]. Available: https://github.com/openstack/tripleo-incubator [retrieved: March, 2015]

[31] Online Documentation for Puppet. [Online]. Available: https://docs.puppetlabs.com [retrieved: March, 2015]

[32] Online Documentation for Ansible. [Online]. Available: http://docs.ansible.com/ [retrieved: March, 2015]

# Time Diversity based Iterative Frequency Estimation with Weighted Average Scheme with Low SNR for Pulse-Doppler Radar

Sangdong Kim
Division of IoT·Robotics
Convergence Research
DGIST(Daegu Gyeongbuk
Institute of Science &
Technology)
Daegu, Korea
kimsd728@dgist.ac.kr

Yeonghwan Ju
Division of IoT·Robotics
Convergence Research
DGIST (Daegu Gyeongbuk
Institute of Science &
Technology)
Daegu, Korea
yhju@dgist.ac.kr

Jonghun Lee
Division of IoT·Robotics
Convergence Research
DGIST (Daegu Gyeongbuk
Institute of Science &
Technology)
Daegu, Korea
jhlee@dgist.ac.kr

***Abstract*— This paper proposes a time diversity based iterative frequency estimation scheme with low signal to noise ratio (SNR) for pulse Doppler radar based on weighted average. The conventional estimator can alleviate the effect of a diffused Doppler frequency. By applying a weighted average scheme to the received signal, the proposed algorithm improves the accuracy of the frequency estimation. The proposed estimator has better performance than the conventional estimators. Not only is the Cramer-Rao lower bound (CRLB) derived but also the performance of the proposed algorithm is analyzed and verified through Monte-Carlo simulations in an additive white Gaussian noise (AWGN).**

***Keywords- time diversity; iterative frequency estimation; weighted average scheme; pulse Doppler radar.***

## I. Introduction

Doppler frequency estimation (DFE) algorithms are related to the estimation of sinusoidal waveforms in various applications such as radar and communication. Maximum likelihood (ML) based estimation [1], discrete Fourier transform (DFT) based algorithms [2][3], adaptive notch filters, the different least mean square error technique [4], and Kalman filters [5] are just some of the most important methods developed for frequency estimations of noisy sinusoids. Furthermore, the estimation of the frequency of a sinusoid is usually performed using a two-step method containing a coarse estimation and a fine search [6][7]. These fine search estimations have variances similar to that of the Cramer-Rao lower bound (CRLB) but nonetheless still show frequency-dependent performance. Iterative frequency estimation (IFE) [7] which is the state-of-the-art in estimation problem allows equal performance to be accomplished independent of the true signal frequency and the estimation accuracy to be improved.

In practical applications, for pulse Doppler radar signals with low signal to noise ratio (SNR), unavoidably, the Doppler frequency is quite diffused and very difficult to estimate. In order to solve the Doppler frequency problem with low SNR, we propose a weighted average for the iterative frequency estimation (WAIFE) which can alleviate the drawbacks to a certain extent.

This paper is organized as follows. The system model of the pulse-Doppler radar is presented in Section II. We describe the proposed WAIFE algorithm in Section III. To validate the proposed algorithm, Monte-Carlo simulations are provided in Section IV. Finally, a conclusion is drawn in Section V.

## II. System Model

We consider the pulse signals used for pulse-Doppler radar [8]. We consider these pulse signals, including the Doppler frequency, in order to extract the velocity of an object. For convenience of explaining the proposed concept, the radar signal is simply assumed to be composed of only *N* pulse repetition intervals (PRIs) during an *M* update time.

The radar signal, composed of only *N* PRI, is simply transmitted such that

$$s(t)=\left\{A_T\cdot\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\exp\left[-2\pi\left(\frac{t-i\cdot T_{PRI}-j\cdot T_{UP}}{\tau_p}\right)^2\right]\right\}\text{for}\quad 0\le t\le(M-1)\cdot T_{UP} \tag{1}$$

where $A_T$ is the amplitude of the transmitted signal, $M$ is the total number of update times, $N$ is the total number of PRIs, $\tau_p$ is a time-normalization factor, $T_{UP}$ is the update time contained in the received signal with $N$ PRI, and $T_{PRI}$ is determined by means of $T_{PRI} = 1/f_{PRI}$.

At this point, the Doppler shift is estimated through an iterative method based on the DFE. The received signals $r_j(t)$ for the $j$-th update time are formulated in terms of the Doppler shift by

$$r_j(t)=\alpha_j s_j(t-\tau_j)e^{j(2\pi f_{d,j}t+\theta_j)}+\omega(t)\,, \tag{2}$$

where $\alpha_j$, $f_{d,j}$, $\tau_j$ and $\theta_j$ represent the complex amplitude, the Doppler frequency, the time delay, and the initial phase affected by moving objects during the $j$-th update time, respectively. In addition, $\omega(t)$ is the complex additive white Gaussian noise (AWGN) with variance $\sigma^2$ .

Assuming that the received signals are sampled at a rate of $f_s = 1/T_s$, the received $N$ ($T_{PRI}=T_s$) samples $r(nT_s)$, $n=0,\ldots,N-1$ are changed into a sinusoid signal of the Doppler frequency. The received samples $r_j[n]$ are expressed as

$$r_j[n] = \alpha_j e^{j2\pi f_{dn,j} n} + \omega[n], \tag{3}$$

where $f_{dn,j}$ is the Doppler frequency normalized during the $j$-th update time and $\omega[n]$ is the sampled value of $\omega(t)$ with $t=nT_s$. We also assume that the baseband signal is sampled at the peak point of $s(t)$. Here, the normalized Doppler frequency $f_{dn,j}$ is defined in terms of the Doppler frequency $f_{d,j}$ such that $f_{dn,j} = f_{d,j} / f_s = v/N$ during the $j$-th update time, where $N$ represents the number of samples. The number $v$ is composed of $k_{p,j}$ and $\delta$, which are the integer part and the fractional part of the $j$-th update time, respectively.

## III. Weighted Average for an Iterative Frequency Estimation (WAIFE)

In this section, we depict how the WAIFE is employed for the DFE of the received signal with time diversity and how the proposed estimator combines the IFE with a weighted-average scheme by considering the power of the received signal.

### A. Iterative Frequency Estimation

From (3), the IFE is achieved in three steps during the $j$-th update time. The integer part $k_{max,j}$ is initially estimated from the index of the periodogram of the received signal; then, by conducting the first iterative scheme, the fractional part $\delta_{j,1}$ is estimated using the values of the periodogram at the appropriate frequencies. In the third step, the estimation of the fractional part $\delta_{j,2}$ is also iteratively achieved from the results of the periodogram using the previous integer $k_{max,j}$ and the fractional part $\delta_{j,1}$. The frequency corresponding to the maximum DFT amplitude coefficient is chosen as a frequency approximation of the ML estimate. We define DFT such that

$$y_j[k] = \sum_{n=0}^{N-1} r_j[n] e^{-j2\pi kn/N} \text{ for } k = 0,1,\ldots,N-1, \tag{4}$$

where the result of $N$ point complex DFT operator, $\mathbf{y}_j = [y_j[0], y_j[1], \ldots, y_j[N-1]]^T$. Following Rife and Boorstyn [10], the first step of the frequency estimate, $\hat{f}_j$ , can be obtained by means of

$$\hat{f}_j = \frac{k_{\max,j}}{N} f_s, \tag{5a}$$

where

$$k_{\max,j} = \arg\max_k \{y_j[k]\}, \tag{5b}$$

The power of the received complex amplitude $\hat{p}_j$ during the $j$-th update time is also estimated as

$$\hat{p}_j = \max\{\left|\mathbf{y}_j\right|^2\}, \tag{6}$$

Through the first iterative scheme, the fractional part $\delta_{j,1}$ of the Doppler frequency can be estimated using the values of the periodogram. We define the modified DFT coefficients, $X_p$, using the equation below.

$$X_p = \sum_{n=0}^{N-1} r_j[n] e^{-2\pi n \frac{k_{\max,j}+p}{N}}, \; p = \pm 0.5. , \tag{7}$$

Based on the literature [7], we combine the input sinusoidal signal with $X_p$ and carry out the necessary manipulations for the case of a noiseless condition. The yield is

$$\hat{\delta}_{j,q} = \frac{1}{2}\mathrm{Re}\left\{\frac{X_{0.5}+X_{-0.5}}{X_{0.5}-X_{-0.5}}\right\}, \tag{8}$$

where the number of iterations $q=1$.

Last, to improve the estimation accuracy, we obtain the fractional part $\hat{\delta}_{j,2}$ using the second iterative scheme. Using the previous $k_{max}$ and $\hat{\delta}_{j,1}$, the modified DFT with $q$=2 is achieved such that

$$X_p = \sum_{n=0}^{N-1} r_j[n] e^{-2\pi n \frac{k_{\max,j}+\hat{\delta}_{j,1}+p}{N}}, \; p = \pm 0.5, \tag{9a}$$

and

$$\hat{\delta}_{j,2} = \hat{\delta}_{j,1} + \frac{1}{2}\mathrm{Re}\left\{\frac{X_{0.5}+X_{-0.5}}{X_{0.5}-X_{-0.5}}\right\}. \tag{9b}$$

In order to obtain the fraction part $\hat{\delta}_{j,a}$ iteratively, we estimate the normalized Doppler frequency during the $k$-th update time using the equation

$$\hat{f}_{dn,j} = \frac{k_{\max,j}+\hat{\delta}_j}{N} f_s, \tag{10}$$

where $\hat{\delta}_j = \hat{\delta}_{j,q}$ such that $q$=2.

### B. Weighted-Average Algorithm

In practical applications with a low SNR, unavoidably, the Doppler frequency cannot be obtained accurately because it is diffused. To deal with this problem, the WAIFE considers the time diversity in the DFE. Through [9], when the Doppler frequency is slowly varied, the WAIFE precisely estimates the smoothed Doppler frequency. From (10), let $\hat{\delta}_j$ ($j$= 0, 1, . . . , $M$-1) and $\hat{p}_j$ denote the fractional part of the Doppler frequency and the received power during the $j$-th update time, respectively. Because the received signal with a good SNR is weighted, the WAIFE shows better performance.

To estimate a frequency that is contaminated with noise, the weighted-average Doppler frequency is expressed using

$$\hat{\delta}_{wa} = \sum_{j=0}^{M-1} \frac{\hat{p}_j \cdot \hat{\delta}_j}{\sum_{i=0}^{M-1} \hat{p}_i} \tag{11}$$

Once the estimated $\hat{\delta}_{wa}$ is obtained, a new Doppler frequency can be constructed using

$$\hat{f}_{dn} = \frac{k_{\max} + \hat{\delta}_{wa}}{N} \qquad (12)$$

## IV. PERFORMANCE ANALYSIS

In this section, the root-mean-square error (RMSE) and the CRLB for the proposed WAIFE are derived. For an arbitrary $f_{dn,j}$ during the $j$-th update time, the RMSE of the iterative algorithm used for DFE was analyzed in earlier work [7] when the number of iterations is one in an AWGN channel. The RMSE of the proposed WAIFE is derived such that

$$\mathrm{var}\left[\hat{\delta}_{wa}\right] = \frac{\pi^2\left(\delta^2 - 0.25\right)^2\left(4\delta^2+1\right)}{4 \cdot M \cdot N \cdot SNR \cdot \cos^2\left(\pi\delta\right)}, \qquad (13)$$

where the received power $\hat{p}_j$ ($j$=0,1,…,$M$-$1$) is constant at one in the $j$-th received signal and where SNR is the average signal-to-noise ratio.

According to the CRLB, the DFE is based on the frequency estimation for the transformed sinusoid of the Doppler frequency. For the Doppler frequency signal of (3) in the AWGN channel from earlier research [10], the CRLB for the DFE is derived such that

$$\delta_f{}^2 \geq \frac{6 f_s{}^2}{2\pi \cdot SNR \cdot N \cdot M \cdot (N^2 - 1)}. \qquad (14)$$

## V. SIMULATION RESULTS

We present Monte-Carlo simulations to assess the Doppler frequency estimation performance of WAIFE. We define the RMSE as $\frac{1}{L}\sum_{n=1}^{L}\sqrt{(\hat{\delta}_{wa} - \delta)^2}$ with $L$=1,000. In the following simulations, we normally adopt a pulse-Doppler radar system as the setup with $M$=5, $f_s$=1, and $N$=1,024, and we assume that there is one moving source. Fig. 1 shows the RMSE values of various instances of $\delta$ in the proposed and in conventional estimators. Here, SNR = 0dB and $N$=1024 is used for practical situations of the proposed and conventional estimators. Compared to the ML estimator and the IFE with $q$=1, the proposed estimator has better RMSE performance than the conventional estimation such as [7] by more than about 40 times. Specifically, the proposed estimator shows performance similar to that of the CRLB at approximately -0.3 < $\delta$ < 0.3.

Fig. 2 presents the performance results for various estimators when $\delta$ =0.3. We compare our algorithm with the IFE and the ML methods. Fig. 2 shows that our algorithm is capable of much better Doppler frequency estimation performance and that the performance is very close to that of the CRLB. The proposed estimator shows that the analyzed and the simulated performances are in good agreement from approximately SNR=-13dB. Fig. 3 shows the RMSE of various estimators according to an increase in the FFT size $N$. When $N$ increases, the RMSEs of the proposed estimator, the ML and the IFE is improved. In particular, with the proposed estimator, when $N$ changes from 256 to 4,096, the RMSE characteristics improve by more than 60 times, with a change from 5.4e-5 to 8.6e-7. From Fig. 1 to Fig. 3, the proposed estimation improves the estimation performance through the strong evaluation compared with the conventional estimation.

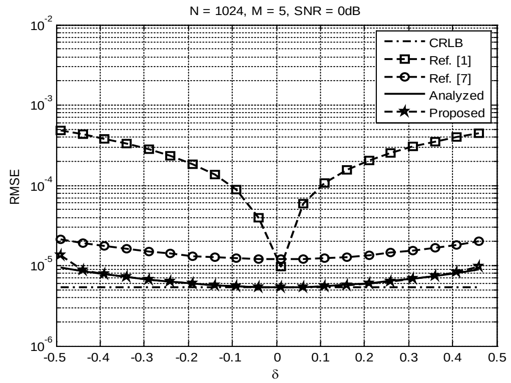


Figure 1. Performance of the WAIFE based on various $\delta$ with SNR = 0dB.

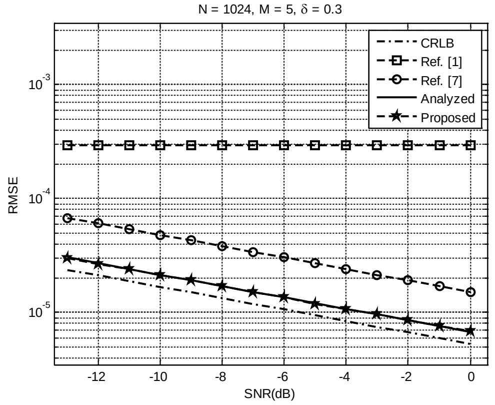


Figure 2. Performance of the WAIFE compared to that of the IFE and the ML methods with $\delta$ = 0.3.

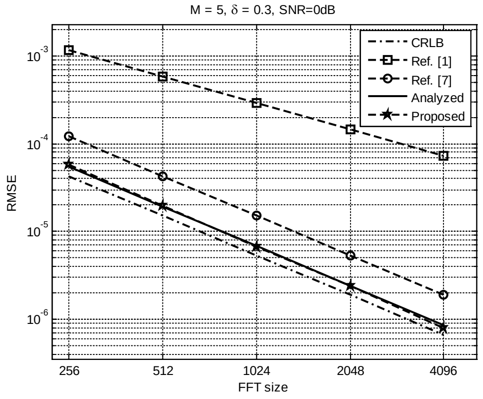


Figure 3. Performance of the WAIFE according to various FFT sizes with SNR = 0dB.

## VI. Conclusions

In this paper, we proposed a WAIFE algorithm which improved the estimation performance of diffused Doppler frequencies in pulse-Doppler radar. Compared to conventional algorithms, our algorithm provided much better performance for the DFE and was capable of performance very close to that of the CRLB. The proposed algorithm worked well with other FFT sizes of $N$. At -0.3< $\delta$ <0.3, the proposed estimator showed performance similar to that of the CRLB. In the future, the proposed estimator can be applied to various surveillance and safety systems based on pulse-Doppler radar that require good performance levels.

## Acknowledgment

This work was supported by the DGIST R&D Program of the Ministry of Science, ICT and Future Planning, Korea (15-RS-01).

## References

[1] R. M. Narayanan and M. Dawood, "Doppler estimation using a coherent ultrawide-band random noise radar," IEEE Transactions on Antennas and Propagation, vol.48, no.6, Jun. 2000, pp. 868-878.

[2] L. Guosui, G. Hong, and S. Weimin, "Development of random signal radars," IEEE Transactions on Aerospace and Electronic Systems, vol.35, no.3, Jul. 1999, pp.770-777.

[3] S. Kim and J. Lee, "A memory-efficient hardware architecture for a pulse Doppler radar vehicle detector," IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences, vol.E94-A, no. 5, May 2011, pp.1210-1213.

[4] E. K. Walton, V. Fillamon, and S. Gunawan, "Use ISAR imaging using UWB noise radar," In Proceedings of the 8th Annual AMPTA Symposium, Seattle, WA, Oct. 1996, pp. 167-171.

[5] I. P. Theron, E. K. Walton, S. Gunawan, and L. Chai, "Ultrawideband radars in the UHF/VHF band, " In Proceedings of the 8th Annual AMPTA Symposium, Seattle, WA, Oct. 1996, pp. 167-171.

[6] C. Candan, "A Method for fine resolution frequency estimation from three DFT samples," IEEE Signal Processing Letters, vol.18, no.6, Jun. 2011, pp.351-354.

[7] E. Aboutanios and B. Mulgrew, "Iterative frequency estimation by interpolation on Fourier coefficients," IEEE Transactions on Signal Processing, vol.53, no.4, Apr. 2005, pp. 1237- 1242.

[8] S. Kim and J. Lee, "A New fine Doppler Frequency Estimator based on Two-Sample FFT for Pulse Doppler Radar," IEICE Transaction on Communications, vol.E96-B, no. 6, Jun. 2013, pp.1643-1646.

[9] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. II. Algorithms and applications," Proceedings of the IEEE, vol.80, no.4, Apr. 1992, pp.540-568.

[10] D. Rife and R. R. Boorstyn, "Single tone parameter estimation from discrete-time observations," IEEE Transactions on Information Theory, vol.20, no.5, Sep. 1974, pp.591-598.

# All-In-One Streams for Content Centric Networks

Marc Mosko and Ignacio Solis

Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304, USA
Email: {mmosko, isolis}@parc.com

***Abstract*—To reduce download latency, All-In-One content streams concatenate many Content Centric Network (CCNx) Content Objects in to one named stream so a requester may open up a large Interest window and download a set of related content objects without incurring a round-trip time to first read a manifest. We describe the structure of an All In One Stream, and how the manifest allows the use of the single object stream or the parallel use of independent streams. We also describe how a modular technology called Reconstructable Objects allows an in-path cache to create sections of the All-In-One content stream from cache rather than having to fetch the entire stream from the origin. Modeling shows improved download speeds of up to 57%.**

***Keywords–Information Centric Networking, Content Centric Networks, Flow Aggregation***

## I. Introduction

As is well known from typical server applications [1], round trip times can add significant latency to content download. For example, with HyperText Transfer Protocol (HTTP) over Transmission Control Protocol (TCP), one typically has a 3-way TCP handshake then a GET request before beginning to download the desired HTTP and HyperText Markup Language (HTML) markup. Then, after parsing the HTML markup, the client can request individual embedded objects. Just as there are efforts to reduce to so-called “zero round trip time” protocols in the IP world, we may devise such protocols for CCNx [2]. All In One Streams is an efficient and secure zero-round trip time protocol that allows a client to skip already cached pieces of the download and to exploit external caching of embedded objects.

In an Information Centric Networks (ICN), such as Content Centric Networks (CCNx) or Named Data Networking (NDN) [3], one uses a flow control algorithm to manage a receiver-driver congestion control protocol (for example [4]–[6]). For our purposes, the exact flow control algorithm is not important, because they generally all share the same overall characteristics. For a given base name, for example /parc.com/slides/allinone.ppt, the content is chunked in to a series of smaller objects with individual names, such as by appending a chunk number to the name. The flow controller then issues requests for names and controls the number of outstanding requests based on its congestion control algorithm, which are variations on additive increase multiplicative decrease (AIMD). This means in the beginning of the download of each base name, the Interest window will be small and then grow. When downloading many related items, such as collateral on a web page, this could lead a flow control algorithm to have multiple slow starts. We propose a method to aggregate multiple items with different base names in to one flow control stream so the same Interest window can be used for them all avoiding multiple slow starts and avoiding stop-and-wait round trips when downloading a table of contents (Manifest) that specifies what other objects are needed.

It is not sufficient for a flow controller to simply open up a large Interest window for each Content Object without knowing how many chunks make up that object. For example, if an image is only 16 KB, perhaps three chunks, and the flow controller asks for 8 chunks, it is wasting 5 Interests. In some systems this is only an upstream bandwidth penalty. However, some systems traffic shape upstream Interests based on expected downlink usage [7]. In such a system, asking for chunks beyond the end of an object will needlessly penalize a consumer and cause delay of other parallel stream.

All-in-one streams is also an efficient mechanism to handle Manifest content. A CCNx Manifest enumerates other content objects that make up the parent object. Normally, one would need to fetch the manifest first, then fetch the constituent pieces. Using an All-In-One stream, the manifest and it contents may be fetch in a zero-round-trip Interest stream. We extend this mechanism to include ways for intermediate caches to serve embedded objects from cache rather than fetching them as part of the stream using a technique we call Reconstructable Objects.

In a typical CCN deployment, a system encodes user data in to Content Objects. Each network transmissible content object contains a small piece of the original data, say 1500 to 8000 bytes. After a receiver has all the constituent objects, it may reconstruct the original user data. If the receiver wishes to re-transmit the content objects, it must save them, along with the user data, because the content objects are cryptographically signed by the original publisher and the receiver cannot reconstitute that signature if it throws away the objects. This leads to a potentially large set of duplicate data on a user’s system to store the content object representation and the original user data representation. We describe a method to avoid the duplicate data using Reconstructable Content Objects. When applied to an All-In-One stream, Reconstructable Objects allows intermediate caches to respond to parts of the stream without needing to fetch the entire stream from the source.

Section II describes how some protocols operate requesting one element at a time. This happens, for example, when one needs to download some or all of a manifest first before downloading the embedded content, causing extra round-trip times. Section III describes our solution for All In One streams that allow a consumer to begin downloading one named stream that contains the concatenation of all pieces. While this scheme avoids extra round trips for manifest download, it does

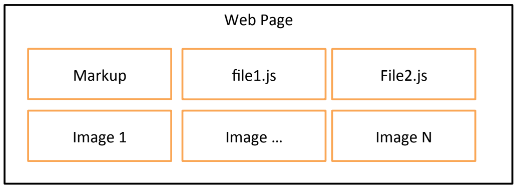


Figure 1. Typical web page components

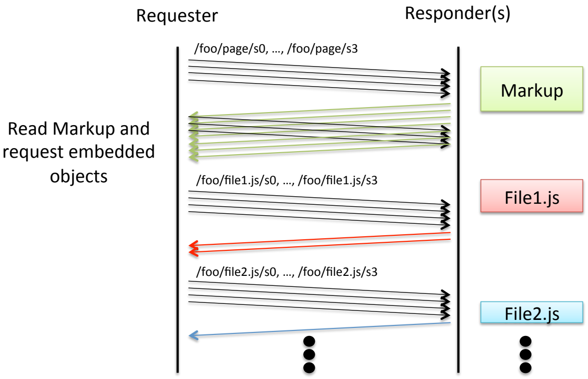

Requester
Responder(s)
/foo/page/s0, …, /foo/page/s3
Markup
Read Markup and request embedded objects
/foo/file1.js/s0, …, /foo/file1.js/s3
File1.js
/foo/file2.js/s0, …, /foo/file2.js/s3
File2.js

Figure 2. One-by-one download

not exploit caching of external collateral, such as images or embedded pages. Section IV describes a method using what we call Reconstructable Objects to allow intermediate caches to contribute external content to the All In One stream, resulting is zero round trips plus the ability for intermediate caches to contribute to the stream. Section V presents performance models of All In One streams versus a round-trip Manifest style system. Section VI concludes the paper.

## II. One-by-one protocols

Conventionally, a CCNx system would issue some initial set of Interest messages to read an piece of content, such as an HTML markup page, then begin downloading the individual components from that page. The download is constrained by some maximum number of outstanding interests that saturate the bottleneck link. Also, a requester does not know *a priori* how many chunks to request, so for small objects the requester may waste some outstanding interests by over-requesting chunks.

For example, in Figure 1, a notional web page consists of the markup and objects references from the markup. These subordinate objects may include some javascript files, such as file1.js and file2.js and embedded images. To download the complete web page, a client must first ask for the markup, and then begin parsing the markup for embedded objects to request those objects.

As illustrated in Figure 2, a conventional download may end up staggering the download of embedded objects because of the need to retrieve the manifest first. Also, because the Requester does not know the number of chunks in an object *a priori*, the Requester may open up too large of a window, such as when the Requester issues 3 interests for file1.js when it only needed to issue 2 interests. The extra interest could have been used to request useful content, such as beginning the download for file2.js.

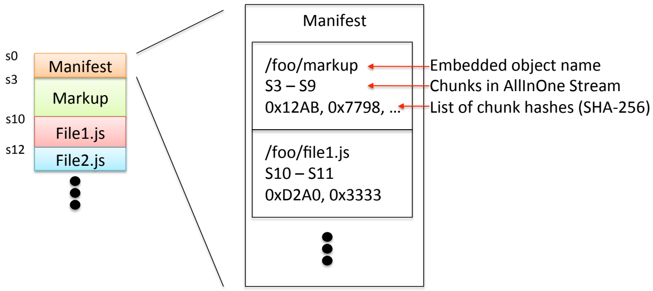


Figure 3. All In One Manifest

```
Stream = Manifest *(Embed)
Manifest = *(ManifestEntry | ManifestContinuation)
ManifestEntry = Name StartChunk ChunkLength
                [Flags] (ChunkLength)(Hash)
Name = <CCNx Name>
StartChunk = ChunkNumber
ChunkLength = Number
ChunkNumber = Number
Flags = [gzip]
gzip = <embedded object gzip'd>
Hash = <SHA-256 ContentObjectHash>
ManifestContinuation = [PreviousChunk] [NextChunk]
PreviousChunk = ChunkNumber
NextChunk = ChunkNumber
```

Figure 4. ABNF for All-In-One Stream

## III. All In One Streams

An All In One Stream is a method for a content provider to aggregate a set of content in to one named stream. A Stream consists of an initial Manifest followed by constituent objects. The entire stream is under one chunked namespace, so a requester can open up one large Interest window that downloads the manifest plus all locally served content. The manifest has enough information that a requester could skip specific embedded objects if it already has them.

Figure 3 shows an example manifest, which is an array of manifest entries. Each entry has the CCNx base name of the embedded object (minus the chunk number), the list of chunk numbers in the All In One stream, and the Content Object Hash of each embedded content object. Enumerating the chunk ranges of each embedded content object allows the requester to determine if the object is already covered by the outstanding Interest window or to skip embedded content objects if the Requester already has an object with the same Content Object Hash. Enumerating the Content Object Hashes of each embedded object allows the requester to open up separate Interest windows for the embedded object and request it by hash.

Figure 4 shows the Augmented Backus-Naur Form (ABNF) for an All-In-One stream. It begins with a stream Manifest and is followed by embedded Content Objects. The stream manifest enumerates the CCNx base name (name before chunk number) of an entry, the starting chunk number in the embedded stream and the chunk length in the embedded stream. It also enumerates the ContentObjectHash of each embedded object (the inner object, not the All-In-One stream object). At minimum, the first content object of an All In One stream should carry the public key used to sign the stream so intermediate nodes and end systems can verify signatures.

The Manifest may be interleaved through the All-In-One stream using ManifestContinuation records which point to the previous Manifest and next Manifest using the All-In-One stream chunk numbering. This organization is similar to a Linear Tape File System (LTFS) [8] tape volume, where Index and Data Extent partitions can be sequenced in the LTFS Volume. Rather than include generation numbers in each index, we use the fact that there is a common CCNx name prefix for each chunk of the All-In-One stream, such as with a version name component, to indicate that all indices and data extents belong to the same stream.

The publisher creating an All-In-One stream may organize the manifest to optimize download. For example, the manifest may list only a few initial content object hashes of each embedded object rather than a complete enumeration. Listing only a few initial hashes gives the requester enough information to determine if they have the object and to begin downloading it under its own namespace. Later index sections could then finish the embedded object enumerations.

The publisher may generate the embedded objects two ways. It could create one embed object per embedded chunk. Or, it could create a non-chunk aligned embedding of the content. Either method may be used. If the stream uses compression then one is generally using the second, non-aligned method. If the gzip flag is present in a Manifest entry, it means the embed object uses the Gzip protocol to compress the inner object.

The Interest request could include a “modified since” parameter in the name, so the All In One stream would only include embedded content objects modified after the embedded parameter. It may also included older objects that are newly referenced, for example if an old photo has not been included in a web page manifest for a long time, it could be included in the All In One stream even though it has not been modified since the Interest parameter.

A Requester may request embedded objects under their own name, using a self-certified Content Object Hash name. Downloading the objects under their own name could allow the download to come from well-positioned caches, whereas downloading the embedded objects might need to come from a less optimal source. For example, image files may have very long cache lifetimes, so would be cached in more places, while the web page might have a short cache lifetime because it is frequently updated.

Figure 5 shows an example All In One exchange, where the producer has concatenated several related assets in to one stream. The Requester may open up an Interest window to begin downloading many chunks. In the figure, for example, the Requester sends 4 Interests, which downloads the Manifest (2 chunks) plus part of the Markup. Once the first chunks return to the Requester, it can open up the window more and continue reading chunks without even parsing and reading the Manifest. This can all happen in a tight control loop of the ICN flow controller.

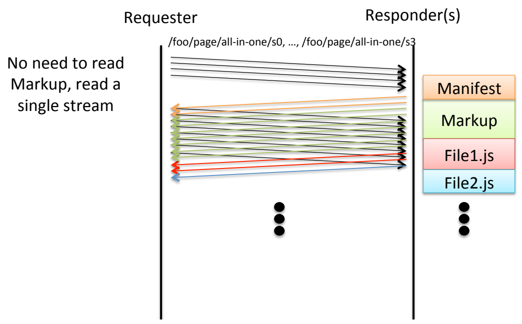

Requester
Responder(s)
/foo/page/all-in-one/s0, ..., /foo/page/all-in-one/s3
No need to read Markup, read a single stream
Manifest
Markup
File1.js
File2.js

Figure 5. All In One Manifest

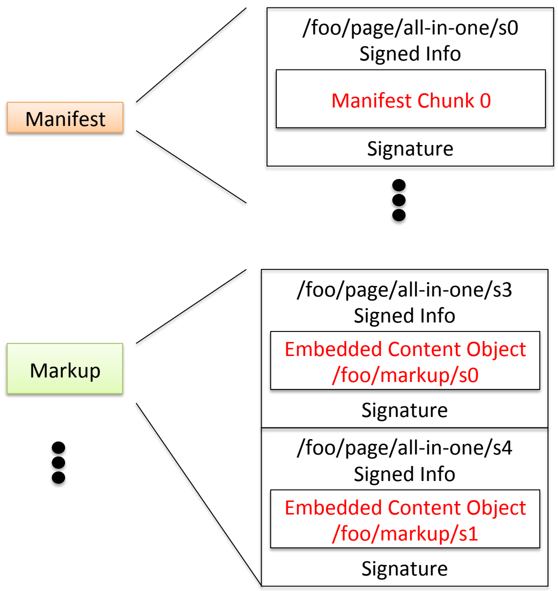


Figure 6. Example All In One Stream Content Objects

Figure 6 illustrates an example stream. The manifest is one or more content objects whose payload is the Manifest. It is signed, as normal, for a CCNx Content Object. The next objects in the stream are embedded inside stream objects. This is because the stream objects have their own stream name. Therefore, /foo/page/all-in-one/s3 is a CCNx Content Object that embeds /foo/markup/s0, being the first chunk of the actual markup. Both the wrapping object and the embedded object have their own CCNx signatures. The embed objects could omit the CCNx signature because of the ability to hash chain from the Manifest.

The embedded objects might themselves be All In One streams. For example, one HTML file might reference frames of other HTML or other objects, which could themselves be organized as an All In One stream. In such a case, the embedded All In One stream is treated like one embedded object in the parent manifest.

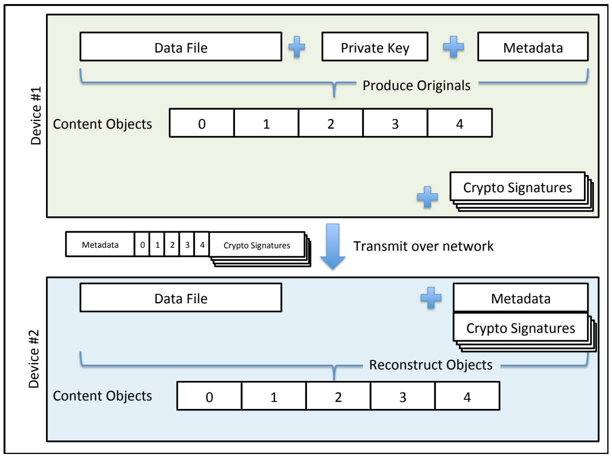


Figure 7. Reconstructing individual content objects

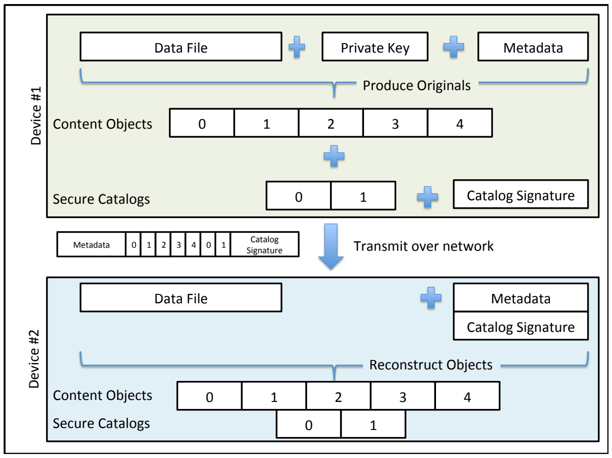


Figure 8. Reconstructing using secure catalog

## IV. All In One Streams and Reconstructable Objects

Reconstructable Content Objects store the original user data as a normal file on a user's system. This is the normal form an application would need to use the data, such as a JPEG or a game's data files. In addition to the user data, the system stores a set of rules. The rules describes how to publish the user data, such as the number of bytes per content object, the timestamps to use, how to name the content objects, and other things that go in each Content Object. With the original data file and the reconstruction rules, a system can verbatim reproduce the original content objects without duplicating the data. Finally, the system stores the original publisher's signature(s) for the data. In a system that uses directly signed content objects, there are many signatures. For a system that uses manifests (secure catalogs), there may only be one or a small number of signatures.

Reconstructable Content Objects consist of several components. There is the underlying user data which is chunked in to one or more Content Objects. The chunking is done via a set of rules embodied in a ruleset. This ruleset provides all information for all content objects except for the cryptographic signatures. Finally, there is a set of cryptographic signatures, one for each signed content object. This is a necessary and sufficient set of objects to construct content objects from only the underlying user data.

Figure 7 illustrates the process of creating and then reconstructing content objects. An initial device, noted as "Device #1" in the figure, has an original data file. Using a metadata ruleset, it constructs and initial set of content objects. The metadata ruleset specifies how a device fills in all the fields of a content object, such as the Creation Time or when to use an End Chunk field, and the format of the CCNx names. The initial device then cryptographically signs each content object. This creates the set of signatures shown in the figure, which are actually part of each content object.

The initial device in Figure 7 then transfers the metadata ruleset plus the content objects over the network to a second device, noted as "Device #2". The second device reconstructs the original user data file and saves the metadata ruleset plus the cryptographic signatures. This this small additional data, it may reconstruct the original content objects by following the metadata ruleset and plugging in the saved cryptographic signatures.

In a variation of this system, the first device may use a secure catalog to authenticate the content objects. This scheme, show in Figure 8, only has a signature on the secure catalog, not on each individual content object. Therefore, the state transferred over the network and saved at the second device is potentially much smaller than the system where each content object is individually signed. In this variation, the rules for creating the secure catalog are in the metadata, so the secure catalog itself does not need to be stores.

### A. All-In-One Streams with Reconstruction

A reconstructable content object comprises a set of rules about how to construct content objects from user data, a set of metadata, a set of cryptographic signatures, and the user data. From this set of data, a system may re-construct a CCNx Content Object without having the actual content object. A reconstructable All-In-One stream uses this system to reconstruct wrapped content objects. The original (inner) content object is the "data file" and the reconstruction rules describe how to wrap the inner content object in the All-In-One stream. This allows, for example, a cache to respond to an All-In-One chunk request while storing only the inner wrapped content object and the reconstruction rules.

A forwarder, for example, may already have an embedded object in its Content Store. Using a reconstructable stream allows the forwarder to reconstruct a stream chunk from the locally cached embedded object. Figure 9 shows an example reconstructable All In One stream. The manifest is now a Constructable Manifest, which includes the normal manifest plus Reconstructable Object rules, metadata, and signatures. This allows a system that already has the embedded object, such as /foo/markup/s0, to construct the All In One stream object, such as /foo/page/all-in-one/s3 without having to fetch the wrapper /foo/page/all-in-one/s3.

An advanced forwarder that understands All In One manifests can pro-actively fetch embedded objects before the user asks for them, then use a the reconstructable stream to pass them on to the user. It could cache the embedded objects

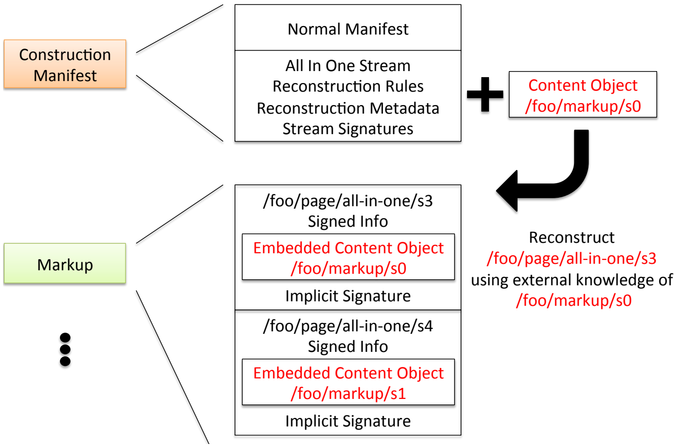


Figure 9. Reconstructing Stream Content Objects

separately from the wrapped stream objects so in the future it could respond with either content object using its hash-based name.

Note that the Construct Manifest also serves as a secure catalog. It contains all the signatures for the All In One stream, so each subsequent content object in the stream may refer to the Construct Manifest for signature verification.

If the Construction Manifest carries a globally unique ID (GUID) (such as a SHA-256 hash of the manifest) in the first content object of the manifest, then an intermediate forwarder could detect if its seen the entire All In One stream before. In such a case, the forwarder can immediately satisfy all interests in the All In One Stream based on cached objects or reconstructable streams. A system should only do this if it can verify the signature of the first content object.

The Construction Manifest could specify the cryptographic hashes of subsequent content objects in the All In One stream. This means the subsequent content objects do not carry any signatures, but are verified solely by the signatures in on the Construction Manifest.

## V. Analysis

We model the performance of All-In-One streams using several stream size distributions assuming a lossless network and TCP-like window behavior. Each stream contains 3, 10, or 30 embedded objects, where each object is uniformly $1KB \dots 50KB$. All content objects are chunked at 1500 bytes (about 1400 bytes of Manifest payload). Only the Manifest chunks are signed. Wrapped objects are not signed, but may be verified based on the Manifest hash. The TCP-like window behavior begins each Content Object download with $K$ initial window Interest packets and doubles until link saturation. We use 10 Mbps as the bottleneck link speed with a total 50 msec one-way latency. We used 5 random simulations per data point and present the average.

We compare a system that uses All-In-One manifests, which maintains a single optimal TCP-like window over the entire stream to a plain Manifest system. In the plain Manifest system, once the first chunk of the Manifest is read, the system can continue reading the referenced objects and we assume the TCP-like window maintains its optimal size. That is, it does not re-start the TCP window for each object. If the flow controller implementation restarts the TCP window per object, the performance will be significantly worse than what we show here. Based on this setup, we expect that the All-In-One stream will show the most improvement for smaller downloads where the advantage of opening up a large initial window gains the most benefit.

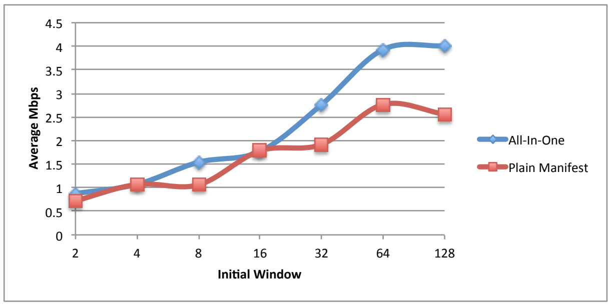


Figure 10. Avearge Mbps for 86 KB download

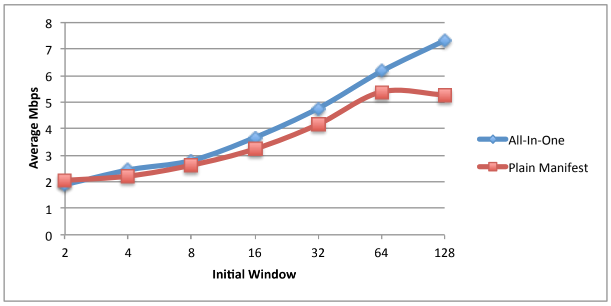


Figure 11. Avearge Mbps for 296 KB download

Figure 10 shows the average Mbps of download speed when the system begins with a given initial window. In these runs the average download size is 86 KB. Even with small initial windows, the All-In-One stream can be 50% faster than the plain Manifest version. At large initial window sizes, the benefit is pronounced and consistent, ranging from a 42% to 57% improvement. At an initial window size of 4, the All-In-One stream was 1% worse than the plain manifest.

Figure 11 shows the average Mbps of download speed where the average download is 296 KB. At an initial window of 2, the All-In-One stream is 8% worse than a plain manifest, and in all other cases, the All-In-One stream is between 7% better up to 39% better with a large initial window.

Figure 12 shows the average Mbps of download speed where the average download is 905 KB. At 2, 4, and 16 for the initial window size, the All-In-One stream was between 2% to 6% worse than a plain manifest. At other values, the All-In-One stream is between 3% to 8% better. Clearly, at this size of download the advantage of eliminating the initial round trip to begin reading data is nearly lost.

## VI. Conclusion

All In One streams is an optimized download method for content networks. It allows a requester to open up one large Interest window and download a manifest plus related objects

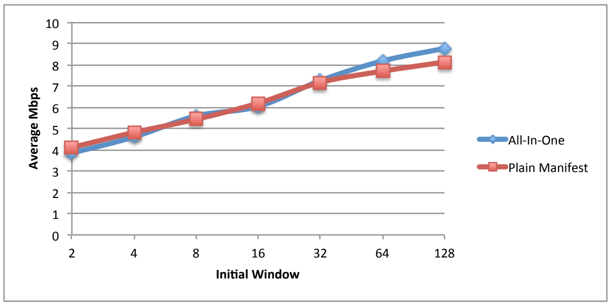


Figure 12. Avearge Mbps for 905 KB download

without needing to parse the manifest before downloading actual content. The organization of an All In One stream allows the requester to skip embedded objects it already has or to download embedded objects from their manifest name using their own namespace.

All In One streams offer several benefits to individual content retrieval. A requester can open up a larger initial Interest window without wasting Interests on many small objects. The embedded content objects can be compressed in the large stream, including the overhead of the names. Using Reconstructable Objects allows caches to answer requests for portions of embedded objects, even though they are being requests under the All In One stream name. The content provider an interleave manifest and embedded content in most any order to optimize the download experience, such as allowing initial rendering of content before all pieces have been downloaded. Finally, All In One streams are secure. Because the initial manifest sections are signed by the provider, all subsequent chunks of the stream can be implicitly validated with their hash.

Modeling of download of an All-In-One stream shows significant improvement, up to 57% faster, than using just a Manifest and waiting for an extra round trip before downloading content. If the download flow controller restarts the download Interest window per object, the improvement of All-In-One streams would be significantly higher. The benefit of All-In-One streams is highest for smaller downloads, say under 500KB, and becomes small by 1MB, where the benefit of having a large initial download window to avoid an extra round trip begins to vanish.

## References

[1] M. Belshe, R. Peon, and M. Thomson, "Hypertext transfer protocol version 2," Working Draft, IETF Secretariat, Internet-Draft draft-ietf-httpbis-http2-17, February 2015. [Online]. Available: {http://www.ietf.org/internet-drafts/draft-ietf-httpbis-http2-17.txt}[accessed:2015-04-10]

[2] "Content centric networking specification," 2015, URL: http://www.ccnx.org/documentation [accessed: 2015-04-10].

[3] "Named data networking," 2015, URL: http://named-data.net [accessed: 2015-04-10].

[4] L. Saino, C. Cocora, and G. Pavlou, "Cctcp: A scalable receiver-driven congestion control protocol for content centric networking," in Communications (ICC), 2013 IEEE International Conference on, June 2013, pp. 3775–3780.

[5] G. Carofiglio, M. Gallo, and L. Muscariello, "Icp: Design and evaluation of an interest control protocol for content-centric networking," in Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on, March 2012, pp. 304–309.

[6] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu, "A transport protocol for content-centric networking with explicit congestion control," in Computer Communication and Networks (ICCCN), 2014 23rd International Conference on, Aug 2014, pp. 1–8.

[7] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An improved hop-by-hop interest shaper for congestion control in named data networking," SIGCOMM Comput. Commun. Rev., vol. 43, no. 4, Aug. 2013, pp. 55–60. [Online]. Available: http://doi.acm.org/10.1145/2534169.2491233

[8] "Linear tape file system (ltfs) format specification," Tech. Rep., August 2011.

# Flexible Design of a Light-Architecture Content Streaming System with Dual Adaptation

Eugen Borcoci, Cristian Cernat, Radu Iorga
University POLITEHNICA of Bucharest
Bucharest, Romania
emails: eugen.borcoci@elcom.pub.ro
cristian.cernat@elcom.pub.ro
radu.iorga@elcom.pub.ro

Jordi Mongay Batalla
National Institute of Telecommunications
Warsaw, Poland
email:jordim@interfree.it
Daniel Negru
LaBRI Lab, University of Bordeaux
Bordeaux, France
email:daniel.negru@labri.fr

***Abstract* —Real time content delivery is a high popularity service in Internet. In contrast with some complex architectures like Content Delivery Networks, Content Oriented Networks, this paper considers a light architecture, working on top of the current networking technologies. It enhances the (video) content delivery via Internet, based on multi-criteria content server initial selection and then in-session media adaptation and/or server handover, all the above in a unified solution. This in-progress work explores different design variants, illustrating the solution flexibility.**

***Keywords — content delivery; multi-criteria decision algorithms; server and path selection; media adaptation, monitoring, Future Internet; content-aware networking.***

## I. Introduction

The content-related services are more and more present in the current and Future Internet, leading to recent significant developments [1]. Dedicated infrastructure like Content Delivery Networks (CDNs) improve the services quality [2], by distributing the content replica to cache servers, located close to groups of users; they are largely used in the real world. Content/Information Oriented/Centric Networking (CON/ICN/CCN) approaches [3][4], decouple names from location, introduce content-based routing, in-network caching, etc. However, all the above need complex architectures, high CAPEX and significant modifications in Service/Content Providers and Network Providers/Operators systems.

As an alternative, Service Providers (SP) might deliver services in *over-the-top* (OTT) style, over the current best effort Internet as a cheaper solution. An OTT SP could be a separate entity from the traditional Internet Service Provider (ISP). Also combined solutions exist, with OTT SPs using the CDN Providers infrastructure to improve the quality of delivery. The OTT solutions approach to improve the quality, when transport problems appear in the network, are frequently based on usage of adaptive solutions for media streams and or servers, in order to maintain acceptable quality at receiver side.

A light architecture (OTT-like), for content streaming systems is proposed by the European *DISEDAN* Chist-Era project [5], (*service and user-based* ***DI****stributed* ***SE****lection of content streaming source and* ***D****ual* ***A****daptatio****N*****,** 2014-2015). The business actors involved are: *Service Provider (*an entity/actor which deliver content services and owns or not a transportation network); *End Users (EU)* which consumes the content; a *Content Provider(CP)* could exist, owning *Content Servers(CS).* DISEDAN does not deal with CP /SP contractual relationships; we assume that servers are owned by the SP. An effective solution is constructed for the multi-criteria hard problem of *best content source (server) selection*, considering user context, servers' availability and requested content. A novel concept is introduced based on: (1) *two-step server selection mechanism* (at SP and at EU) using algorithms that consider context- and content-awareness and (2) *dual adaptation mechanism,* consisting in *media flow adaptation* and/or content source adaptation (by *streaming server switching*) if quality degradation is observed during the media session. Such an OTT-like solution is attractive since it avoids the complexity of CON/ICN or CDN.

*This work is mainly dedicated to analyze the design decisions variants. Details on server/path selection, optimization algorithms and adaptation process combined with server switching are treated in other works [13[[14]..*

The system can be flexibly implemented in several variants, depending on the complexity/constraints envisaged and the EUs and SPs requirements. We explore different design decisions and trade-offs, versus the cost and implementation complexity. This work is preliminary; currently, the system is under its implementation.

Section II is a short overview of related work. Section III outlines the overall architecture. Section IV contains the paper main contributions, analyzing various design decisions and implementation-related implications. Section V contains conclusions and future work outline.

## II. Related Work

Adaptation techniques enhance the quality of streaming media at the consumer side when the transfer conditions deteriorate. It also support efficient network resource utilization, device-independent universal media access and optimized Quality of Experience (QoE). Many Service providers apply it, to solve the network variations [6]. Adaptation may act on Media (flow) [6][7][8], and/or on Content server. The latter means in-session new server selection and switching (handover), depending on the

ISBN: 978-1-61208-404-6

consumer device capabilities, consumer location and/or network state [9][10].

Recent solutions for media adaptation use the HTTP protocol, minimizing server processing power and being video codec agnostic [11]. Relevant examples are: Adobe Dynamic Streaming, Apple's HTTP Adaptive Live Streaming and Microsoft's IIS Smooth Streaming and open HTTP-based protocols like Dynamic Adaptive Streaming over HTTP (DASH) [8]. The DASH continuously select, on-the-fly, the highest possible video representation quality that ensures smooth playout in the current downloading conditions. The DISEDAN novelty [5], conists in "dual adaptation" by combining in a single solution the initial server selection (result of cooperation between SP and EU) and in-session dual adaptation,.

The initial server selection is based on optimization algorithms like *Multi-Criteria Decision Algorithms (MCDA)* [12] modified to be applied to DISEDAN context [13]14], or *Evolutionary Multi-objective Optimization algorithm* [15]. In these works several scenarios are proposed, analyzed and evaluated. In particular, the availability of different static and/or dynamic input parameters for optimization algorithms is considered. Therefore several designs are possible, different in terms of performance and complexity. It is the objective of this paper to analyze these variants, seen as design / implementation decisions.

## III. DISEDAN System Architecture and design guidelines

While considering the above general concepts, assumptions and requirements should be identified, to provide inputs for the system design.

### A. General framework and assumptions

The main business entities/ actors have been mentioned above: SP, EU, CS. The connectivity between CSs and EU Terminals (EUT) are assured by traditional Internet Services Providers (ISP) / Network Providers (NP) - operators. Due to its OTT-style, DISEDAN does not consider SP - ISP/NPs relationships in its management architecture. Some Service Level Agreements (SLAs) might exist, related to connectivity services, but they are not directly visible at our system level. The DISEDAN solution is also applicable to other business models, e.g., involving CPs, CDN providers, etc. The relationships between SP and such entities could exist, but their realization is out of scope of this study.

The system works on top of current TCP/IP mono and/or multi-domain network environment. The EUTs might not have explicit knowledge about the managed/non-managed characteristics of the connectivity services. Network level resources reservation, or in-network connectivity services differentiation are not mandatory supposed (but not forbidden). This shows the system flexibility: it can work in OTT style, or over a managed connectivity services. Therefore, the SP cannot offer strong QoS guarantees to EUs. Consequently, DISEDAN does not manage (but does not exclude) possible EUs/SPs SLA relationships. However, it is assumed that a Media Description Server exists, managed by SP, to which EUT will directly interact.

The media streaming operations are independent on networking technology. The client-side streaming system, acts as a standalone application, (no mandatory modifications for SP); however, SP should provide some basic information to EUT, to help its initial server selection. The decision about dual adaptation are taken mainly locally at EUT, thus avoiding complex EUT-SP signaling.

In a general case, several CSs exist (containing replicas of media objects), known by SP (geo-location, availability, access conditions for users), among which the SP and/or EUTs can operate servers selection and/or switching. No restriction is imposed either on the geo-localization of EUTs or of CSs. Note that the proposed system does not consider how to solve network failures , except attempts to perform media flow DASH adaptation or CS switching. The terminal devices are supposed to have all the required subsystems and peripherals for video/ audio display and device control.

### B. End User Requirements

These requirements are expressed as EU needs, and are related to *user scenarios* - when selecting and consuming media content related services.

- The system should admit the usual user profiles. EU should be able to identify itself and login into the system through a controlled environment.
- The EU should be able to select among several SPs and among content items, servers and classes of quality – in the limits offered by the selected SP.
- The DISEDAN system should allow to EU: initial (optionally automatic or manual) server selection; in-session dual adaptation will be automatically applied, to maximize the Quality of Experience (QoE).
- The EU should receive information from SP (on servers and possibly on network) to help him in selection. The EU should also have the possibility to finally decide on server selection/switching or amount of adaptation actions initiated and/or performed.
- The EUT should be still able to work by using only minimal information on server and network (e.g., server capacity or download bandwidth from the server) delivered by the SP. The selection is basically locally taken, thus avoiding the in-session signaling between user and SP.
- The EU should have the possibility to be informed about of QoE level delivered by the system.
- The client SW installed on the EUT should have maximum independence from the operating system running on the terminal.

### C. Service Provider Requirements

These requirements are expressed as SP business and technical needs. The DISEDAN system:

- Should allow SP to develop multimedia content-based services, e.g., live streamed IPTV services, Video on Demand (VoD) and its derivatives (e.g. streamed VoD, downloaded / pushed content).

- Must allow SP to filter the control information delivered to the EUs, but should not impose major architectural modification in the SP Management and Control (M&C) architecture.
- Should allow SP to apply different policies in its server selection (e.g. to maximize CS utilization).
- Should be able to use the SP static/dynamic (monitored) information on servers and network paths status and availability, in mono or multi-domain contexts.
- Must not restrict the networking technologies (QoS capable or not) used by SP.
- Must support the SP-EU cooperation for dual adaptation purposes.
- Should offer to the SP the minimal capabilities to manage the Content Severs (if no distinct Content Provider business entity exists).

### *D. General System Requirements*

These are results from the previous requirements for: User and Service Provider. The DISEDAN system:

- Must work in the traditional TCP/IP mono and multi-domain, in OTT style, on top of arbitrary network technology; the EUTs or CSs can be placed everywhere.
- Should provide a simple management with minimal architectural modifications at SP side and EUT.
- Must optimize multi-criteria content source selection, and then dynamic dual adaptation, considering user context, servers availability, network conditions and distribution mode. It will apply: a. *two-step server selection* (at SP and EU) based on context/content-aware algorithms; b. *dual adaptation,* (media adaptation and/or server switching).
- At EU side, a standalone client application exists. No mandatory modifications at SP M&C side are required; however SP M&C should provide information to EUT, to help it in initial server selection.
- Should provide flexible possibilities to assign/balance the decision power between SP/EU, regarding sever selection/switching and dynamic adaptation.

Other specific EUT, SP and CS system requirements have been derived from the general ones but they are not detailed here.

### *E. General Architecture*

Figure 1 shows the general architecture. The Service Provider entity includes the following functional modules:

- *MD File generator* – dynamically generates Media Description (MD) XML file, containing media segments information (video resolution, bit rates etc.), ranked list of recommended CSs and possibly - CSs current state information and even information on network state (if applicable).
- *CS Selection (step 1) algorithm* - it exploits MCDA or EMO, to rank the CSs and media representations, aiming to optimize servers' load and to maximize the system utilization.
- *Monitoring module* – collects information from Content Servers and estimates their current states.

The End User Terminal entity includes the modules:

- *DASH* – parses the MD file received from SP and handles the download of media segments from Content Servers.
- *Content Source Selection and Adaptation engine* – implements the dual adaptation mechanism.
- *Selection (*step 2) *algorithm*. It can also exploit MCDA, EMO, or other algorithms to select the best CS from the list recommended by SP.
- *Monitoring module* – monitors the local network conditions and –possibly- the server condtions.
- *Media Player* – playbacks the media segments.

The Content Server entity includes the modules:

- *Streaming module* – sends media segments requested by End Users.
- *Monitoring probe* – monitors CS performance (CPU utilization, network interfaces utilization, etc.). In a complex implementation of the CS, the monitoring probe could be replaced by a more capable monitoring module, to supervise both the active sessions and some connectivity characteristics to different groups of users.

The following (macro) functional steps are:
1. The EUT issues a media file request to SP.
2.The SP analyzes the status of the CSs and runs the selection algorithm (optionally the SP could make first, a current probing of the CSs).
3. The SP returns a candidate CS list to EUT.
4. The EUT performs the final CS selection and starts asking segments from the selected CS.
5. During media session, the EUT measures the quality and evaluates the context. It applies DASH adaptation or if necessary, CS switching is decided.

When the user requests a Multimedia content, the SP sends an *xml* file containing Media Description (MD). This file is updated (from the static *xml* file) for each user request by considering the user profile, the SP policies for this user's class and other information at the SP side (e.g., state of the servers and possibly network-related information). The list of candidate CSs and other information is written inside the *xml* file. Also caching server *url* addresses can be added. The list may be optionally ordered, following some desired metrics. When the user's application receives the MD file, it performs the final CS selection and possibly the network path. This decision can be based on user context or, one can simply select the first CS in the ordered list. The CS selection achieves multi-objective optimization. After final selection, the EUT starts to ask segments from the selected CS. During the receipt of consecutive chunks, the user's application can automatically change the rate of the content stream (DASH actions) or, if still problems exist, it can switch the CS.

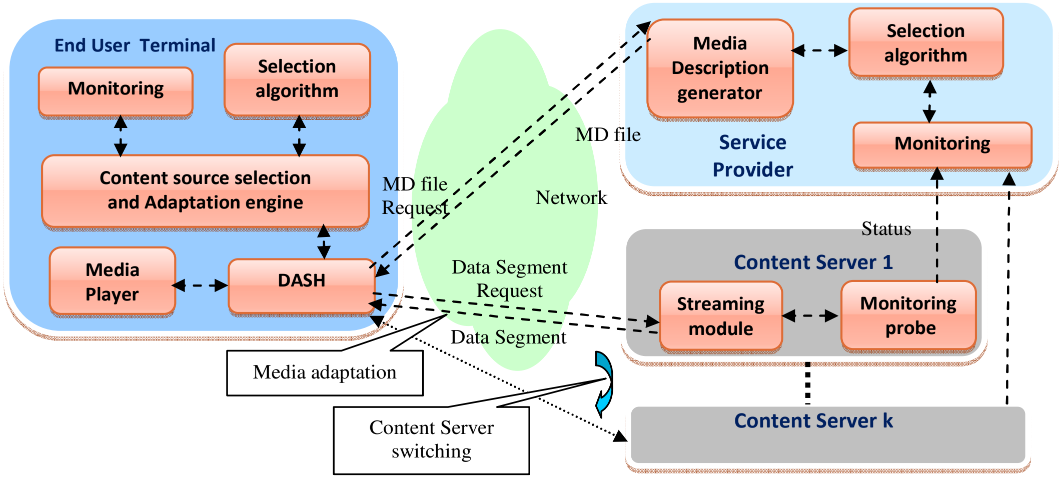


Figure 1. DISEDAN general architecture; DASH - Dynamic Adaptive Streaming over HTTP

The EUT performs measurements on the parameters of download process. If EUT detects deterioration of downloading rate, it can use SP information about alternate CSs and/or it can start probing CSs. After probing, the EUT decides on media or server adaptation.

## IV. Design Decision Variants

The DISEDAN architecture is flexible. Several variants/versions of designs can be considered, i.e., basic ones or more complex, essentially depending on the roles of the business entities and their capabilities, interactions and also on SP and EU policies.

### A. *Monitoring procedures*

The types and amount of static and dynamic monitoring data collected by various entities have a significant impact on the solutions. Consequently, the MCDA/EMO algorithms will have different sets of input parameters. Apart from static information available at SP, three types of monitoring contexts and possible policies can be identified.

*Proactive monitoring:* data are continuously collected (at SP level and possibly at EUT level); they guide the initial CS selection, when some new content requests arrive. At SP, this means supervision of different CSs, maybe network paths and user communities, depending on its policies. SP/CS cooperation on this purpose is envisaged. Such data can be used to construct a history and updated status of the environment envisaged by the SP. The CSs could be involved in proactive monitoring, provided they are capable to probe the connectivity characteristics towards different groups of users (indicated by the SP).

At EU side, in some more complex scenarios, the EUT can construct history, e.g., dedicated to its favorite content connections, or focused on network context.

*Session-time (Real-time) monitoring:* the data are measured on the media flow, basically at EUT. In more complex DISEDAN variants, the SP and/or some CSs can be involved in such monitoring, at least in being aware of results (note that no SLA concerning mutual obligations of SP/EUs, related to QoE exist) for: all active users or subsets; all monitored data or summaries of them; full or summary monitored values.

*Opportunity related monitoring:* these are measurements essentially performed by the EUT, to test the opportunity of switching the CS that delivers the content to EU.

### B. *Possible Roles of the Business Actors*

The DISEDAN project outlines a set of optional Provider side modifications (w.r.t. useful information and metrics provided by SP to the client) that can further optimize server selection. The design can be backwards-compatible, ensuring that each modified client or SP can cooperate with the other side, if the latter is using existing content distribution solutions. Consequently, we propose a range of solutions regarding the SP, CS, EUT roles, i.e., several variants (named “use cases”) and listed below.

The Tables I, II, III illustrate different design choices, listed in increasing order of complexity and consequently of performances, for SP, CS and EUT. Note that, although the Monitoring subsystems could be included generally in the architectural Management Plane, the *Mon@SP*, *Mon@CS* or *Mon@EUT* are specified in the tables in a distinct way, in order to emphasize the dynamic character of the data collected. Depending on the specific requirements and constraints, different variants can be selected as design/implementation choices of the DISEDAN system.

## V. Conclusion and Future Work

This paper presented an analysis of design decisions for implementation variants of a novel and flexible light-architecture content delivery system, working on top of the current Internet networks. The system involves a Service Provider, End Users and Content Servers owned by the SP. The novelty consists in including in a single solution of initial content server selection, (based on collaboration SP - EU, and multi-criteria optimization algorithms like MCDA, EMO, etc.) and session-time DASH adaptation and/or intelligent server switching, if the quality of the flow is degraded at the End User. Several versions of designs, are proposed illustrating the system flexibility and comments are given on the associated complexity.

TABLE I. SP-RELATED DESIGN VERSIONS

| | Information known about: | Obtained from | Type | Monitoring system involved | Remarks on SP role |
|---|---|---|---|---|---|
| **SP-V1** | CS list and locations | Mgmt@SP | Quasi-static | No | SP solves the user requests.<br>SP is involved in initial server selection, or during session (to help switching decision at EUT) , based only on ordered list of servers, depending on their load.<br>(*minimum complexity*) |
| | Content files mapping on servers | Mgmt@SP | Quasi-Static/dynamic | No | |
| | CS status (current load) | CSs | Dynamic | Yes | |
| | User groups | Mgmt@SP | Quasi-static | No | |
| | Active users | EUs | Dynamic | No/yes | |
| **SP-V2** | **Idem as SP-V1, plus below items** | | | | Idem as in SP-V1 but more qualified assistance in selection of the initial (server-path).<br>Problem: how can a given user invoke usage of a selected path if multiple paths are available. |
| | Potential user groups | Mgmt@SP | Quasi-static | No | |
| | Basic connectivity paths static characteristics (at overlay level) from different CSs to different groups of users | Mgmt@SP/ CSs | Quasi-static | No | |
| **SP-V3** | **Idem as SP-V2, plus below items** | | | | Idem as in SP-V2 but more assistance in selection of the initial (server-path), given the paths current load information. |
| | Current loads of the paths (bandwidth availability) | CSs | Dynamic | Yes | |
| **SP-V4** | **Idem as SP-V3, plus below items** | | | | Idem as in SP-V3 but more assistance in selection of the initial (server-path). |
| | Other dynamic paths characteristics (delay, loss, jitter, etc.) | CSs | Dynamic | Yes | |
| **SP-V5** | **Idem as SP-V4, plus below items** | | | | Idem as in SP-V4, but more flexibility from business point of view. |
| | Policy Information | Mgmt@SP | Static | No | |
| **SP-V6** | **Idem as SP-V5, plus below items** | | | | Idem as in SP-V5, plus more powerful set of knowledge on system history.<br>*( maximum complexity)* |
| | Historical and prediction data on servers and paths utilization | Mgmt@SP<br>+ Mon@SP | Dynamic | Yes | |

TABLE II. CONTENT SERVER –RELATED DESIGN VERSIONS

| | Information known about: | Obtained from | Type | Mon@CS involved | Remarks on CS role |
|---|---|---|---|---|---|
| **CS-V1** | EU authorization data | Mgmt@SP | Quasi-static | No | CS solves the user content requests.<br>CS status info is delivered to SP.<br>CS info on active users can be also delivered to SP. |
| | EU requests | EUTs | Dynamic | Yes | |
| | CS status (current load) | Mgmt@CS | Dynamic | Yes | |
| | Active users | EUs | Dynamic | Yes | |
| **CS-V2** | **Idem as CS-V1, plus below items** | | | | Idem as in CS-V1 but more assistance in offering (via SP) additional information for selection of the initial (server-path). |
| | Potential user groups | SP | Quasi-static | No | |
| | Basic connectivity paths static characteristics (evaluated at overlay level) from different CSs to different groups of users | Mon@CSs | Quasi-static | Yes | |
| **CS-V3** | **Idem as CS-V2, plus below items** | | | | These data can be help SP for more efficient management of EU connections. If multiple paths are available, the CSs should have some source routing capabilities in order to force the stream to follow a given path. |
| | Active User groups | Mgmt@CS | Dynamic | Yes | |
| | Connectivity paths dynamic characteristics (evaluated at overlay level) from different CSs to different groups of users | Mon@CS | Dynamic | Yes | |

TABLE III. END USER TERMINAL –RELATED DESIGN VERSIONS

| | **Information known about:** | **Obtained from** | **Type** | **Mon@EUT involved** | **Remarks on EUT role** |
|---|---|---|---|---|---|
| **EUT-V1** | EUT local static context | Mgmt@EUT | Quasi-static | **No** | EUT issues content requests to SP.<br>For server selection it uses the MD file sent by SP and its static context information.<br>For dual adaptation it uses the monitored data and basic probing information. |
| | MD file | SP | Dynamic | No | |
| | QoE quality during session | Mon@EUT | Dynamic | Yes | |
| | CS accessibility (basic probing) | Mon@EUT | Dynamic | Yes | |
| **EUT-V2** | **Idem as EUT-V1, plus items below** | | | | EUT issues content requests to SP.<br>For server selection it uses the MD file sent by SP and its static context information.<br>For dual adaptation it uses the monitored data and probing information. |
| | EUT local dynamic context | Mon@EUT | Dynamic | Yes | |
| | CS accessibility (advanced probing) | Mon@EUT | Dynamic | Yes | |
| **EUT-V3** | **Idem as EUT-V2, plus items below** | | | | |
| | Local Policy information | Mgmt@EUT | Static | No | Possible local policy data are used in server selection and dual adaptation. |
| **EUT-V4** | **Idem as EUT-V3, plus items below** | | | | |
| | Historical and prediction data on servers and paths utilization | Mgmt@EUT<br>Mon@EUT | Dynamic | Yes | Possible history and prediction data are used in server selection and dual adaptation. |

A main DISEDAN adavantage consists in avoiding to develop complex M&C planes and signaling, while still offering sufficient QoE (due to adaptation capabilities ) to the end users, in a cheap and fast implementable OTT-style solution.

Simulations have been performed, including large scale network environment, to prove the capabilities of the proposed architecture. Partial results assessing the validity of the solution and performance of the algorithms are already reported in [13][14]. Ongoing work is currently performed, to implement the described system (in the DISEDAN project). Performance of the implemented system, obtained for different use cases, will be reported in some future papers.

## ACKNOWLEDGMENTS

This work has been partially supported by the Research Project DISEDAN, No.3-CHIST-ERA C3N, 2014- 2015.

## REFERENCES

[1] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures", IEEE Communications Magazine, vol. 49, no. 7, , pp. 26-36, July 2011.

[2] P. A. Khan and B. Rajkumar. "A Taxonomy and Survey of Content Delivery Networks". Department of Computer Science and Software Engineering, University of Melbourne. Australia : s.n., 2008. www.cloudbus.org/reports/CDN-Taxonomy.pdf.

[3] ***, "Information-Centric Networking-3", Dagstuhl Seminar, July 13-16 2014, Available from: http://www.dagstuhl.de/en/program/calendar/semhp/?seminar=14291

[4] V. Jacobson, et al., "Networking Named Content," CoNEXT '09, New York, NY, pp. 1–12, 2009,

[5] http://wp2.tele.pw.edu.pl/disedan/publications

[6] T. Dreier, "Netflix sees cost savings in MPEG DASH adoption," 15 December 2011. [Online]. Available from: http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=79409. [Accessed October 2014].

[7] S. Wenger, Y. Wang, T. Schierl and A. Eleftheriadis, "RTP Payload Format for Scalable Video Coding," IETF RFC 6190, 2011.

[8] ISO/IEC 23009-1, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats," ISO/IEC, Geneva, 2014.

[9] M. Kawarasaki, K. Ooto, T. Nakanishi and H. Suzuki, "Metadata driven seamless content handover in ubiquitous environment," in Proceedings of the 2004 International Symposium on Applications and the Internet SAINT'04, Tokyo, 2004.

[10] S. Park and S. Jeong, "Mobile IPTV: Approaches, Challenges, Standards and QoS Support," IEEE Internet Computing, vol. 13, no. 3, p. 23–31, 2009.

[11] L. De Cicco, S. Mascolo and V. Palmisano, "Feedback control for adaptive live video streaming," in MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems, San Jose, California, 2011.

[12] J. Figueira, S. Greco, and M. Ehrgott, "Multiple Criteria Decision Analysis: state of the art surveys", Kluwer Academic Publishers, 2005.

[13] A. Beben, J. M. Batalla, W. Chai, and J. Sliwinski, "Multi-criteria decision algorithms for efficient content delivery in content networks", Annals of Telecommunications, vol. 68, Issue 3, pp. 153-165, Springer, 2013,

[14] E.Borcoci, M.Vochin, M.Constantinescu, J. M. Batalla, D.Negru, "On Server and Path Selection Algorithms and Policies in a light Content-Aware Networking Architecture", ICSNC 2014 Conference, Available from: http://www.iaria.org/conferences2015/ICSNC15.html

[15] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach". IEEE Transactions on Evolutionary Computation, No. 3(4),pp. 257-271, November 1999.

# Video Distribution by Special Hardware Tools

More events now need 4K video for projection in high quality and with low latency

Sven Ubik, Jiri Navratil, Jiri Halak
Research department for network applications
CESNET a.l.o.
Prague, Czech Republic Country
ubik@cesnet.cz, jiri@cesnet.cz, halak@cesnet.cz

Jiri Melnikov
CESNET / Faculty of Information Technology
CTU at Prague
Prague, Czech Republic
Jiri.Melnikov@fit.cvut.cz

***Abstract*— This paper describes the technical options and the use cases of the 4K Gateway technology for specialized distributions of various forms of HD, 4K and 3D videos for high quality projections on large screens. 4K Gateway can be used for transmissions of medical, cultural or sports events or for scientific collaboration. It is based on an FPGA design, which adds minimal internal latency (from 3 ms). It allows to transmit up to eight video channels in both directions. Based on available bandwidth, it can work in an uncompressed or compressed mode. For compression it uses a JPEG 2000 encoding and decoding core.**

***Keywords*— HD Video; 4K video; 3D technology; cyber performance; live surgery.***

## I. INTRODUCTION

Today, downloading a YouTube video usually takes just seconds. Videos are available in several quality levels, therefore almost any type of network can be used. But there is a significant difference if we want to see a video recorded for fun by a simple web camera or if we want to observe a video during complicated surgeries done by endoscopic tools, via microscopes or via the da Vinci Surgical System with special 3D cameras located inside the patient body. It is different when we transmit a cultural event in real time to a remote site where several hundreds of people are in a big hall. In such cases, the video resolution and the capacity of the network are critical parameters. For medical applications the highest possible resolution is desirable HD (High Definition) or 4K (4096 x 2160) formats. Also if we want to project the video with wide angle projectors on walls several meters wide and high, the highest resolution of 4K or 8K is needed otherwise the picture will be blurred.

For live transmissions for real-time collaboration, we need to minimize the end-to-end delay. Therefore, we use uncompressed transmissions, which require significant network bandwidth. To calculate the required bitrate for the transmission we should take into account the video resolution, frames per second and other parameters such as color encoding and possible compression. The resulting bandwidth for one full HD (1920 x 1080) channel can range from several Mb/s to approx. 1.5 Gb/s for the uncompressed transmission.

## II. TRANSMITTING TECHNOLOGY

For the transmission of real time video from the source place (surgery theatre, concert hall, etc.) into the conference venue or into the theatre we need an end-to-end connection. A shared Internet or dedicated links can be used. Sometimes it is a combination of both methods. Particularly, the last segments to venues usually need to be upgraded by a dedicated connection installed for the event.

Video transmission can be done in several ways. Videoconferencing tools or streaming technology (e.g., using RTMP – Real Time Message Protocol) will deliver the video to anybody who will connect to an MCU (Multipoint Control Unit) or a streaming server. Current commercial videoconferencing devices use mostly the H.323 protocol family and heavy compression so that they can work over inexpensive links with minimal bandwidth. The required bandwidth for an HD channel is in the range from 128 Kb/s to 6 Mb/s, depending on the compression level. In our experience, 4 Mb/s is the lowest bandwidth for acceptable quality for medical transmissions. We have used them for a couple of years for transmissions of ophthalmological surgeries.

For Internet streaming, we usually use hardware H.264 encoders (e.g., Makito) to contribute the picture to a remote streaming server (such as Wowza). A web page then allows multiple users to individually connect using one of several distribution formats (FLASH video, HTML5) and resolutions. These encoders are small, very easy to use and compatible with common streaming servers. On the other hand, they introduce a very long latency, 5-10 seconds are not uncommon. Therefore, they are not suitable for side channels for the interaction with the surgeon.

One of the first uncompressed tools was DVTS (Digital Video Transport System) [1,4] developed in Japan. DVTS accepts image sources (e.g., digital camcorders, surgical instruments) over the IEEE 1394 interface and streams them over an IP network. The quality of the picture was mostly defined by the digital camera and lighting conditions. The system was heavily used in medicine. For example in 2011 we broadcasted via DVTSplus a neurosurgery executed by prof. Takanori Fukushima during his visit at Masaryk Hospital in Usti nad Labem to many Asian partners, see Figure 1. The necessary bandwidth for this system was 30-35

ISBN: 978-1-61208-404-6

Mb/s per channel including audio. This was only a minor problem, especially in an academic environment where plenty of bandwidth is available.

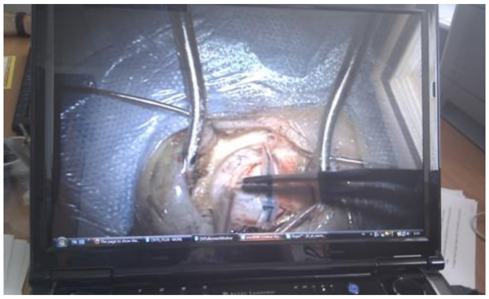


Figure 1. Neurosurgery image via DVTS

## III. 4K GATEWAY

For low-latency real time transmissions of multi-channel video signals CESNET has developed an FPGA-accelerated solution called 4K Gateway [5]. The interface to the video input and output is through HD/3G-SDI interfaces and the interface to the network is by an interchangeable optical XFP transceiver. All video processing is implemented inside an FPGA (Field-Programmable Gate Array) that is by a specialized programmable hardware. This allows to eliminate all varying delays a PC platform causes by data copying, interrupts, device drivers, etc. The video processing is done by a sequence of firmware modules. Communication with firmware is through registers, which can be accessed from within an embedded Linux operating system, which runs on a Microblaze soft-core processor inside the FPGA.

Firmware can operate in uncompressed or JPEG 2000 modes. In the uncompressed mode, it can send and receive up to eight HD signals simultaneously, or up to two 4K signals. The processing latency on the sending and receiving device combined is as low as 3 ms. For higher resiliency to network jitter, additional buffering can be configured, at the expense of added latency. In the JPEG 2000 mode, the firmware can send or receive up to four HD signals simultaneously, or one 4K signal. The processing latency is currently 5 frames on the sending and receiving devices combined. Subframe latency based on multiple tile compression will be available in the future. The bitrate needed to transmit one HD channel depends on the frame rate, color encoding, whether embedded audio and ancillary data are transmitted and the level of compression. For example, for 1080p60 4:2:2 10-bit, the uncompressed video bit rate without packet overhead is 1920x1080 pixels x 60 frames x 2 color samples x 10 bits = 2.48 Gb/s. With JPEG 2000 compression, the bitrate for one HD channel is reduced to 20-100 Mbps depending on the quality requirements, with the higher end providing sufficient quality even for the most critical applications. Selected outputs at the receiving side can be synchronized at one-pixel precision, which allows high-quality 3D projections. This functionality allows to combine several resources (different cameras and digital video devices) and to create environment for perfect illusion of presence in a surgery theatre, concert stage or in the sport stadium.

An important property of the 4K Gateway device is the ability to maintain the sender - receiver synchronization without GPS or a similar timebase signal and keeping low latency. There are two sources of rate difference between the data arriving to the receiver from the network and the data that needs to be sent to the rendering device.

First, the internal clock of the sender can be different from the internal clock of the receiver, within a tolerance permitted by the respective transmission protocol formats. For example, the HD-SDI clock rate is specified as 1.485 Gb/s or 1.485/1.001 Gb/s with 100 ppm tolerance, while the 3G-SDI clock is twice this frequency.

Second, jitter can be introduced due to network traffic conditions. This network jitter needs to be accommodated by the receiver FIFO (First In First Out) memory.

Several alternative methods can be used to compensate the data rate difference: receiver feedback, frame buffer, blank period adjustments or rendering clock adjustments.

The receiver can send feedback to the sender requesting sending rate adjustments. This technique is used in window-based transport layer protocols, such as TCP or in some link layer protocols, such as PAUSE frames in Ethernet.

This can be too slow reaction and may require a large receiver FIFO memory, which would introduce high added latency. However, the main problem with this technique is that it requires a cooperating sender. The technique would not work with current cameras and other real-time video sources.

Frame buffer requires the receiver to have a FIFO memory large enough to accommodate several complete frames. Then the rendering device can be driven by a fixed clock oscillator in the receiver. After certain time when the skew between the sender and receiver clock rates causes the frame buffer to overflow of underflow, a frame skipping or duplication is used. In the worst case when both the sender and receiver clocks are shifted by 100 ppm in the opposite directions, the error can expand to the whole frame in 1/200 * $10^{-6}$ = 5000 frames. At 60 frames per seconds it is just 1.5 minutes. A large FIFO memory can extend this time at the cost of added latency.

A precise external clock source can be used to guarantee that the sender and receiver clocks are in sync. GPS receivers are commonly used for this purpose. A limitation of this method is that it is often difficult to get the GPS signal through the building to the sender or receiver location.

We use a method of rendering clock adjustments [7]. Adjusting the clock in HD-SDI channels between the receiver and the rendering device within the permitted tolerance gives the receiver some level of adaptation to the rate of incoming data. This solution requires tunable oscillators and a closed-loop controller in the receiver. In order to adjust the rendering clock to the data source rate, we used a common PID controller (proportional-integral-derivative controler). The complete receiver control structure is shown in Fig. 2.

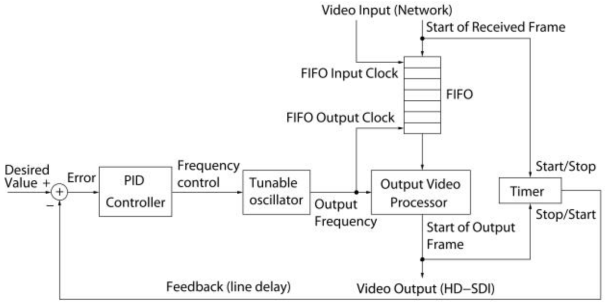


Figure 2. Rendering clock adjustment

The FIFO input is driven by data arriving from the network. The FIFO output is driven by a clock generator, which is tuned by the PID controller. The FIFO occupancy is used as a controlled variable. The desired value is set for required jitter accommodation and latency acceptance. The feedback value is taken from the current FIFO occupancy, which is sampled by the regulator every 200 ms. The controller then uses a weighted moving average of eight last samples. The purpose of this average is to smooth out fluctuations in FIFO occupancy due to network jitter. The PID controller then produces adjustments to the clock generator frequency.

Pairs of channels can be used for 3D signals and multiple channels can be used for tiles of beyond-high-definition signals. A frame detector looks for the beginning of frames in each specified group. After synchronization is applied, the frame generator is temporarily stopped until all channels in a group are aligned.

The device includes multiple tunable oscillators along with their own control loops as in Fig. 2. This allows to flexibly configure all SDI outputs into groups, where each group is independently adjusting its speed to the sender. In this way, one device with eight SDI outputs can receive and present several multi-channel signals in parallel, for example, one 4K signal, one 3D HD signal and two independent HD signals.

## IV. TRANSMISSION EXAMPLES

We have provided multiple telepresence events, with applications mostly from the medical and cultural fields. The events were focused on demonstrations of new possibilities when using fast international research networks, such as GLIF (Global Lambda Integrated Facility) [2,3].

In 2010, CESNET started intensive collaboration with the Robotic Center of Masaryk Hospital in Usti nad Labem. This is one of the few hospitals in Czech Republic equipped with the robotic daVinci Surgical System, which uses a 3D camera to provide the surgeon a stereoscopic vision of the surgery area.

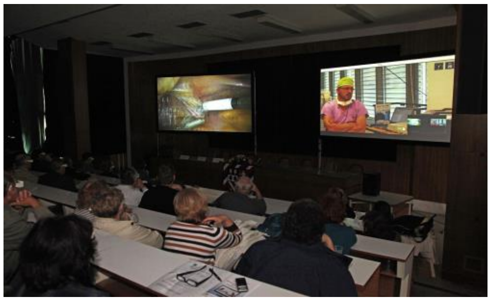


Figure 3. 3D medical transmission from the daVinci Surgical System

In Fig. 3 we show an example of a 3D medical transmission [6] from Usti nad Labem to the hospital in Banska Bystrica in Slovakia. We used this technology for transmissions also to far destinations, to the KEK research institute in Tsukuba, Japan, to the APAN meeting in Nantou, Taiwan and to the Internet2 workshop in Denver, USA. For 3D transmissions, we used a portable ProjectionDesign projector with active glasses, which can use an ordinary projection screen and therefore does not require any arrangements from the meeting organizer. For 4K transmissions we used 4K TVs, which are now available from $500 (as of 2014).

It appears that Internet surgery streaming to individual doctors will be on rise, which can be accelerated by the increasing popularity of mobile devices, such as tablets and increasing mobile bandwidth, with the arrival of LTE (Long Term Evolution) services.

However, we believe that physical meetings at symposia bring additional value of direct information exchange among doctors. What turned very interesting for us with the technical background, is the stimulating role of medical symposia on technical requirements for surgery transmissions. The sheer fact that the doctors devote some of their time to come together results in the flood of new requirements of what else they want to see during the event to maximize the use of time. The first requirement is now to see multiple surgeries in parallel, performed at different institutions. This allows to ”skip” the less interesting parts and concentrate on the more interesting steps when they happen. The second requirement is to use multiple screens, usually one 3D screen switched to the surgery with the currently most interesting phase and several 2D screens for surgeon commentary, surgery room view, and other surgeries.

An example of a multi-screen event was the 19th Congress of the Slovak Society of Gynaecology and Obstetrics in Bratislava, Slovakia. We provided a real-time 3D transmission of the robot-assisted hysterectomy with bilateral adnexectomy and systematic pelvic lymphadenectomy performed by MUDr. Tibor Bielik, CSc. at the Faculty Hospital in Banska Bystrica. The surgery appearance in the lecture hall is shown in Fig. 4. The moderator was commenting on the currently most interesting phase of particular operations.

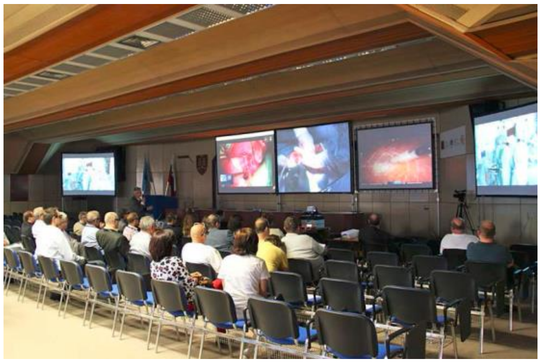


Fig. 4. Telesurgery from multiple hospitals to a congress in Bratislava

In Fig. 5 we show how real-time collaboration among countries can result in a cultural cyber performance. In the culture field, we have demonstrated interactions for remote control of models to access the national cultural heritage, we presented several chamber concerts in 3D vision or concerts with remotely playing musicians together with local performers.

This particular performance was named ”Dancing beyond Time” and involved approx. 100 people in three continents. The final event took place at the 36th APAN Meeting held in Daejeon, Korea on 21 Aug 2013. The event began at 08:55 UTC/GMT simultaneously in Salvador, Brazil (BR), Prague, Czech Republic (CZ), Barcelona, Spain (ES) and Daejeon, Korea (KR). The team included network engineers and researchers, audio-visual technicians, programmers, musicians, dancers, scene designers and choreographers, with some people spanning multiple areas. The music performance was captured by a 4K camera and delivered from HAMU to KAIST by a pair of FPGA-based 4K Gateway devices, which also provided a backward HD channel from KR to CZ for stage monitoring. Audio channels were transferred embedded in the video channel, which guaranteed a perfect video to audio sync in KR. The 4K video was sent uncompressed to preserve high quality. The bitrate was approx. 5 Gb/s. The audience could thus observe a collaborative work of artists physically distributed in different continents.

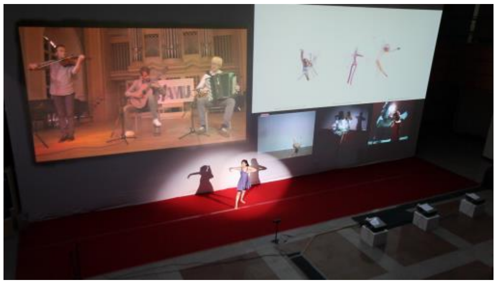


Figure 5. Four-countries, three-continents real-time collaboration in musical and dance performance

## V. Conclusion

The 4K Gateway was originally designed for 4K video contribution. Due to its very low added latency, it can be used for remote access to scientific visualizations, for medical sessions connecting operating theatres with lecture halls and conferences venues and for eCulture events and collaboration. It has been successfully used in various applications, which need high quality and low latency transmissions. In the future we plan to investigate the use of immersive visualizations for collaboration in performing arts, such as the CAVE devices (Cave Automatic Virtual Environment), involving other kinds of artistic expressions, such as fine arts and paintings and installations of more permanent infrastructures for the use in university lectures.

## Acknowledgment

This work was supported by the CESNET Large Infrastructure project (LM201005) funded by the Ministry of Education, Youth and Sports of the Czech Republic and used multiple academic networks including Geant, TEIN, GLIF, Gloriad, KREONET, TWAREN and Internet2.

## References

[1] A. Ogawa, K. Kobayashi, K. Sugiura, O. Nakamura, J. Murai, *Design* and Implementation of DV based video over RTP, IEEE Packet Video Workshop, Cagliari, Italy, 2000.

[2] Tom DeFanti, Cees de Laat, Joe Mambretti, Kees Neggers, Bill St.Arnaud. TransLight: a global-scale LambdaGrid for e-science, Communications of the ACM, Vol. 46, No. 11, Nov. 2003, pp. 34-41.

[3] GLIF: Linking the world with light, Informational brochure, http://www.glif.is/publications/info/brochure.pdf.

[4] S. Shimizu, K. Okamura, N. Nakashima, Y. Kitamura, N. Torata, Y. Antoku, T. Yamashita, T. Yamanokuchi, S. Kuwahara and M. Tanaka, High-Quality Telemedicine Using Digital Video Transport System over Global Research and Education Network, Advances in Telemedicine:

[5] J. Halak, S. Ubik. “MTPP - Modular Traffic Processing Platform”, *Proceedings* of the IEEE Symposium on Design and Diagnostics of Electronic *Circuits and Systems*, 2009, Liberec, pp. 170-173.

[6] J. Navratil, M. Sarek, S. Ubik, J. Halak, P. Zejdl, P. Peciva, J. Schraml, Real-time stereoscopic streaming of robotic surgeries, Healthcom 2011, Columbia, Missouri, USA.

[7] J. Halak, M. Krsek, S. Ubik, P. Zejdl, F. Nevrela, Real-time long-distance transfer of uncompressed 4K video for remote collaboration, FGCS 27(7): 886-892, 2011.

# Synthesis of Two Solutions of Mobility Prediction Based on Data Mining Techniques

Linda Chamek
Laboratory LARI
Informatique Departement
UMMTO
Tizi-Ouzou, Algeria
chameklinda@yahoo.fr

Mehammed Daoui
Laboratory LARI
Informatique Departement
UMMTO
Tizi-Ouzou, Algeria
mdaouidz@yahoo.fr

Malika Belkadi
Laboratory LARI
Informatique Departement
UMMTO
Tizi-Ouzou, Algeria
Belkadi_dz@yahoo.fr

Mustapha Lalam
Laboratory LARI
Informatique Departement
UMMTO
Tizi-Ouzou, Algeria
lalamustapha@yahoo.fr

***Abstract*— In this paper, we present a synthesis of two solutions for mobility prediction using data mining techniques such as classification and clustering. Our solution can be implemented in a third generation network by exploiting user information (age, function, residence place, work place, etc.), the existent infrastructure (roads, etc.) and the history of displacements. Simulations carried out using a realistic model of movements showed that our strategy can accurately predict up to of 80% of mobiles' displacements and this by not knowing their history of mobility**.

***Keywords - mobile network; prediction; data mining; location management.***

## I. INTRODUCTION

Nowadays, mobile networks have become an integral part of our daily life. Third generation networks open the way with new demands for services in multi-media and real time applications. These applications require more communication resources and higher Quality of Service (QoS) than traditional applications. However, these networks are confronted with various problems including resources wasting and signal attenuation that reduce the QoS. User mobility also generates QoS degradation and the network must deal with it.

Two functions are essential in mobile networks: location management and resource reservation. The location management locates the cell where a mobile user is in order to make a call to him. The resources reservation is intended to ensure continuity of communication when a mobile moves from one cell to another by reserving bandwidth in the cells he goes through.

The mobility of users causes performance degradation in relation to the two previous functions. For the localization, the network sends location messages (page messages) to all cells in order to locate the mobile. These messages consume a part of the bandwidth. For the reservation of resources, the network is often required to reserve resources in cells that the mobile will not cross.

Now, if the network would have information about the displacements of every mobile, and if it integrates intelligent strategies to take advantage of this information, it can anticipate their future movements with high accuracy. This way, the network can better manage its resources.

In this work, we present two solutions for mobility prediction based on data mining techniques: classification and clustering. The rest of the paper is organized as follows. In Section II, we present a state of the art of different techniques of prediction. In Section III, we present the importance of the use of data mining in mobility prediction. In Section IV, we propose a prediction solution and, in the last section, we present the evaluation of the solution and the simulation results.

## II. STATE OF THE ART

Several techniques allowing prediction have already been discussed.

One of the most used techniques rests on the localization by Global Positioning system (GPS). The mobile sends its position obtained by GPS to its base station. The latter determines if the mobile is at the edge of its cell. At each reception of the position from the mobile, the system calculates the distance separating the mobile from the neighboring cells, and the shortest (near) distance is selected [1].

L. Hsu *et al.* [2] suggest a solution based on the definition of a reservation threshold. The idea is to compare the signal received by the mobile coming from the neighboring cells. If this signal is lower than the threshold, it is concluded that the mobile moves towards this cell. In [3], a map of signal power is maintained by the system. It represents the various signals recorded in various points of the cell. The authors use this map to know the position of the mobile, and to extrapolate his future position.

In [4], [5], mobility rules are generated based on a history of the movements that each mobile built and maintained during its displacement. These rules are used in the prediction process. In fact, it was observed that users tend to have a routine behavior. Knowing that, and knowing the usual behavior of the users, it becomes possible to predict the next cell which a user will visit.

Soh et al. [6] propose a technique based on the use of a multi-layer neural network in order to exploit the history of the mobile movements. The recent movements of the mobile are initially collected in order to know in which Location Area (LA) it is. A mobility model of the users is initially processed, and then it will be injected into the neural network. The mobility model represents the mobile movements history recorded in an interval of time. The movement is defined in terms of the direction taken and the distance covered.

The role of the neural network is to capture the unknown relation between the last values and future values of the mobility model; that is necessary for the prediction.

The authors in [7] propose an algorithm composed of three phases. The first phase consists in extracting the movements of the mobile to discover the regularities of the inter-cellular movements; it is the mobility model of the mobile. Motility rules are extracted from the preceding model in the 2nd phase. Finally, in the 3rd phase, the prediction of mobility is accomplished by using these rules.

Capka *et al.* [8] propose a new mobility prediction algorithm. The user's behavior is represented by the repetition of some models of elementary movements. To estimate the future position of a mobile, the authors propose an aggressive mobility management Predictive Mobility Management (PMM). A whole of prevision algorithms MMP (Mobile Motion Prediction) is used to predict the next position of the mobiles based on their movements' history.

The authors in [9] propose a diagram of prediction combining between two levels of prediction: global and local. The global mobility model Global Mobility Management (GMM) is given in terms of cells crossed by a mobile during its connection time. The local model Local Mobility Management (LMM) is given by using sample of 3-tuples taking into account three parameters: speed, direction and position. LMM is used to model the intra-cellular movements of a mobile, whereas the GMM is used for the inter-cellular movements by associating its current trajectory with the one of the existing mobility rules. However, the authors do not present any method allowing the discovery of these mobility rules.

A method called Dynamic Clustering based Prediction (DCP) is presented in [10]. It is used to discover the mobility model of the users from a collection containing their trajectories. These rules are then used for the prediction. The trajectories of the users are grouped according to their similarities.

Daoui *et al.* [11][12] present a technique of prediction based on the modeling of mobile displacements by an ants system. This model allows the prediction based on old displacements of the mobile and those of the other users who go in the same direction.

Chamek et al. [13] use a technique of prediction based on classification of users according to their personal profile (age, sex, place of work, etc). A new user is compared to all other users in cell and put in a class, then the history of displacements of the users in this class is used for predicting the next cell of the new user.

A technique based on clustering is presented by Belkadi *et al.* in [14]. This technique can be implemented on next generation mobile networks by exploiting the data available on the users (age, function, address, workplace, etc), existing infrastructures (roads, location of base station, etc.) and the users' displacements history. Locations areas are formed according to these different pieces of information.

## III. DATA MINING AND MOBILITY PREDICTION

The mobile's displacements are often generated by socio-economic needs and are governed by the topography of the roads and infrastructures covered by the various cells of the network such as: schools, factories, supermarkets, highways, etc. The displacements related to the socio-economic needs are usual, and consequently, represent a regular aspect [15].

Information concerning a user characteristics, in other words, the profile, are also of great importance. In fact, knowing certain characteristics of a user helps us to know his future displacements with a great probability. For example, a person of an age ranging between 18 and 25 years, who is student, will probably be located in the campus one day of week. People having high incomes will most probably make their purchases in luxury shops, contrary to others who will prefer supermarkets. The profiles of mobility of these people are thus different.

Many definitions of data mining can be found, so this domain is the subject of research. Engineers, statisticians, economists, etc., can have different ideas on what this term means. We retain a definition which seems to make the compromise between various designs. We can define data mining as the process allowing the extraction of predictive latent information from wide database [16].

Classification aims to predict the class of a new user based on the class of users who are in data base. In our case, it is to predict the future cell of a user based on his last displacements and those of other users who have the same profile. The basis of this idea is that displacements of users are often regular and individuals of the same profile perform similar movements.

Clustering allows to group cells in clusters, thus forming a location area, to facilitate the search of the mobile in the network.

## IV. PRESENTATION OF THE SOLUTIONS

Assume we have an architecture of third generation network composed of a set of cells. Every cell is generated by a base station. The base stations are connected to the core network wired backbone (Figure 1). We assume that the core network has personal and professional information about users such as age, marital status, occupation etc. This information may be collected when subscribing to network services. We also assume that each base station has a history of movements of mobile users.

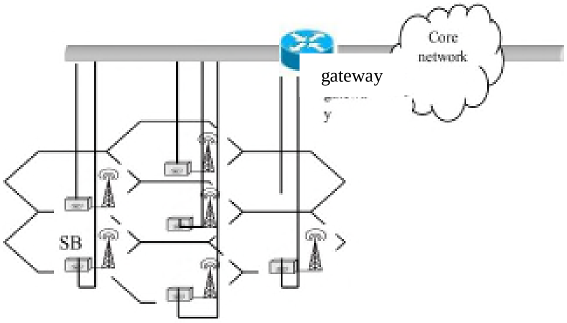


Figure 1. Architecture of a third generation mobile network

This history contains the mobile user id, the cell from which he came (Source cell), the cell to which he is moved (Destination cell) and the date of travel (Table I). This information can be retrieved in the connections log files at each base station and stored in a database on the station itself.

TABLE I. STRUCTURE OF A HISTORY LINE

| ID Mobile | Source Cell | Destination Cell | Date |
|---|---|---|---|

### A. Principle of the prediction technique based on classification

Classification is used to predict the future cell of a mobile user x. We select N mobile users who are already located in the same cell. We compare the user x to every individual y of N users using distance D [12]:

$$D(X,Y)=\sqrt{\sum_{i=1}^{m}(x_i-y_i)^2} \qquad (1)$$

where:

- $x_i$ and $y_i$ are values of attributes of individuals X, Y (like age, function, etc.)
- m is the number of known attributes of individual

So, we select K individuals nearest to X. Then, we take the displacement history of these K individuals. The destination cell which is the most frequented is considered the future cell prediction.

The algorithm proposed is summarized as follows:

**Let** I= {Y1, Y2 ,…. Yn} be the set of the N individuals being in cell C.

Entry : Let X be a new individual for whom we want to predict the future cell.

***Parameter K*** corresponds to the number of nearest neighbors to take into account.

***Parameter L*** corresponds to the number of history lines to take for each nearest neighbor.

Exit: the future cell to predict.

```
Algorithm:
For (j = 1 to N) Do
    1. Calculate the distance between Yj and X of the cell C d(X, Yj)
    2. Record this distance in the vector tab
Done
3. Sort the calculated distances (the vector)
4. Select K smaller distances,
5. Select the history of K individuals closest to X
6. Determine the most frequent destination cell and return it as the future cell to predict
```

### B. Principle of the prediction technique based on clustering

Clustering can be used to form location areas. It is a set of cells in which a mobile can as a function of its history and profile. We define the distance between two cells according to an individual X [13]:

$$D(C_1,C_2)=\sqrt{(n_1-n_2)^2} \qquad (2)$$

with:

$$n_i=\frac{\text{Number of apparition of individual X in the cell i}}{\text{Total number of apparition s of the individual X}}$$

Then, we apply the k-mean algorithm [14] to select K homogeneous clusters. Such cluster contains a set of cells having as a common factor the frequency of visit of a mobile. The cluster having the higher number of visits is considered the appropriate location area for the mobile.

### C. Localisation

The location procedure (paging) that we propose uses an intelligent paging resting on the location areas that are built by the clustering algorithm, such that:

- The first zone of localization is composed of two cells:
  * The cell in which the mobile was at the time of its last call
  * The predicted cell, by our algorithm of prediction, starting from the cell in which the mobile was at the time of its last call.
- The second zone is composed of the adjacent cells to the last cell in which the mobile was at the time of its last call by excluding the cells of the first zone of localization.
- The third zone of localization is composed of the other cells of the network.

**Localization Procedure**:

- Search the mobile in the first location zone
- **if** the mobile is not in the first zone **then**
    - Search it in the second location zone
- **if** the mobile is not there **then** search it in the third location zone, etc.

## V. ADJUSTMENT AND EVALUATION

Most data mining algorithms require a training phase to adjust their parameters. In case of the classification, it provides the best value of two parameter K (number of nearest neighbor to take into account) and L (number of line of history to take by close neighbor).

For clustering, we found the best value of parameter M which represents the optimal number of location zone to create for the network.

In the study of the mobility management and in the absence of a real trace of mobiles displacement, we can resort to a model. The choice of a realistic mobility model is essential.

This model reproduces, in a realistic way, displacements of a set of users within the network. The majority of works presented in the literature use probabilistic models (Markov model, poisson process, etc.) which generate either highly random displacements or highly deterministic displacements which do not reflect the real behavior of the mobile users.

In our approach, we have chosen the activity model presented in [18]. This model is based on the work carried out by planning organizations and uses statistics drawn from five years of surveys on user displacements. It simulates a set of user displacements during a number of days. The generated displacements are based on each user’s activity (work, study, etc.), the locations of these activities (house, work places, schools) as well as the ways which lead to these locations.

The simulator rests on the statistics of displacement led in the area of Waterloo [17] and recorded in the form of matrix called activity matrix indicating the probability of arrival of an activity and duration matrix indicating the probability that an activity takes a given period of time. These statistics, as well as information concerning the users, such as the profile (full time employee, student, part-time employee, etc) and the infrastructures (roads, trade, stadium, etc) are recorded in the simulator database.

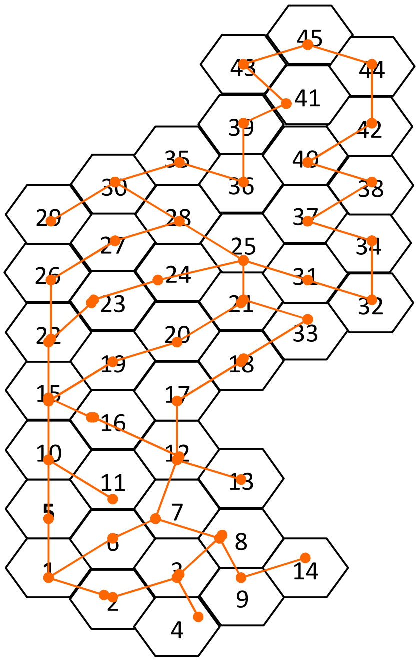


Figure 2. Cellular structure of the simulator

The area of Waterloo is divided into 45 cells, as indicated in Figure 2. According to the activity of the user, the simulator generates an activity event for a user based on the activity matrix and assigns it to a cell. It generates then the displacements relative to this activity before generating the following activity. The process continues until the end of the simulation.

### *D. Adjustment and evaluation of the classification algorithm*

The adjustment of the classification algorithm consists of determining optimal values for the two parameters K and L. K is the optimal number of neighbors to consider, and L is the number of history lines to take for each near individual. The evaluation is done based on rate prediction which is the ratio of the number of correct predictions to the total number of attempts to predict.

Figure 3 shows the ratio prediction as a function of the parameter K, with the L value fixed to 4 (take 4 lines of history for each neighbor). The prediction ratio rises to stabilize at K = 30 with ratio prediction of 60%.

Figure 4 gives the ratio prediction as a function of the parameter L when K is fixed to 30 (the value that we get above). The prediction ratio rises with the rise of the value of L. A better ratio is obtained for L= 45 with a ratio of 70%. So, we can only keep the last 45 displacements of neighbors' users.

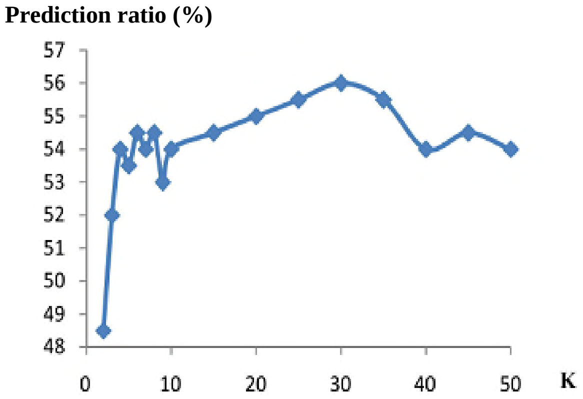


Figure 3. Prediction ratio according to parameter K (optimal number of neighbor)

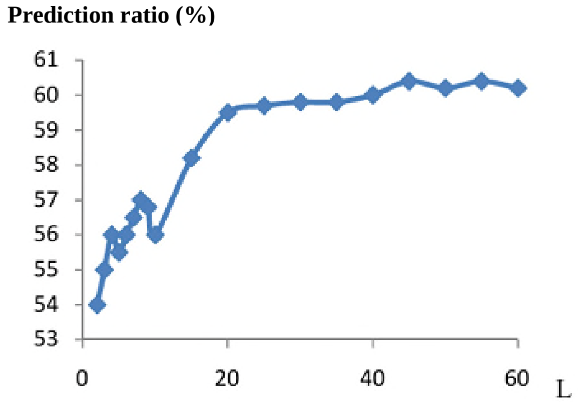


Figure 4. Prediction ration according to parameter L (number of history lines to take for each near individual)

### *E. Adjustment and evaluation of the clustering algorithm*

The adjustment of clustering consists in determining the value of M corresponding to the optimal number of location zones that we need to create for the network. The evaluation is based on the number of paging messages and the update ones. For 100 days of simulation, we have varied M and calculated the total number of messages (paging and update). The results are shown in Figure 5. The optimal number of location zone is 20 with 2 to 3 cells per zone.

ISBN: 978-1-61208-404-6

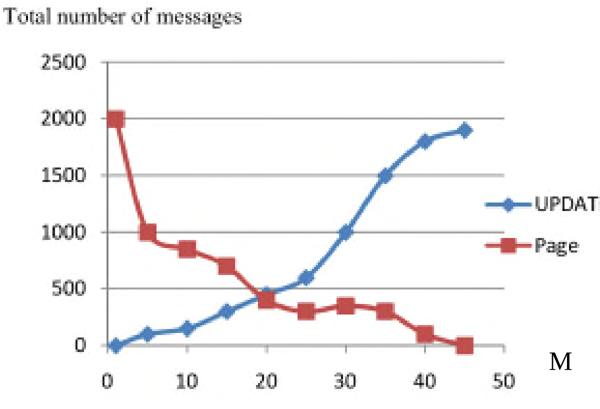


Figure 5. Page messages and updates ones in function to M (optimal number of cluster)

Next, we compared our algorithm with several algorithms, both static and dynamic. In the static strategies, cells are grouped on static location areas (0, 1, 2 and 3). The strategy static 0 contains a cell in each location area. In static 1, 13 locations areas are created, each one having 3 or 4 cells. The static 3 divides the network in 5 areas of 8 to 10 cells. In the dynamic strategy, the algorithm defined in [17] is used. The result of this comparison is illustrated in Figure 6 and Figure 7: the number of update messages and page messages are calculated during 50 days of simulation with 3, 9 and 12 calls per day.

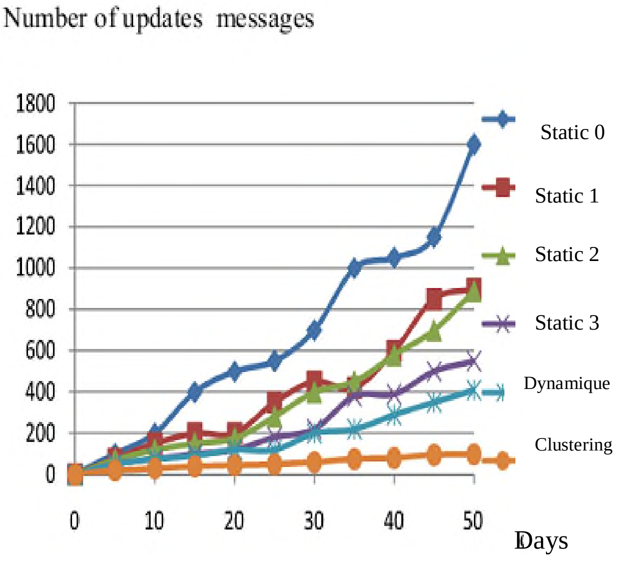


Figure 6. Number of Update messages for 6 strategies in function of days

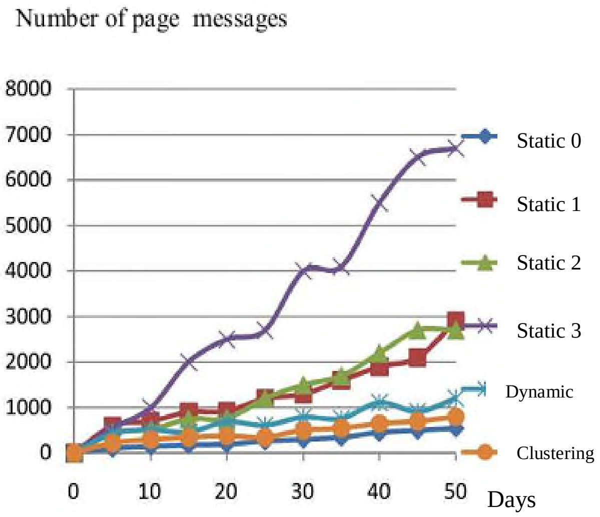


Figure 7. Number of page messages for 6 strategies in function of days

The results show that our solution produces the minimum number of page messages comparing with the other strategies.

## VI. CONCLUSION

Human displacements are often caused by socio-professional needs. They are linked to existing infrastructure (roads, transport, workplace location, etc.). It is therefore possible to predict the future position by looking for links between these movements and other available information such as user profiles and the location of the infrastructure.

Due to the complexity of the characteristics of human mobility and the absence of reliable mobility rules for prediction movements, data mining can be a solution to the problem of prediction. Both techniques presented in this paper show that it is possible to predict 70% of the movements of mobile users.

## REFERENCES

[1] W. Zhuang, K.C. Chua and S. M. Jiang, Measurement-Based Dynamic Bandwidth Reservation Scheme for Handoff in Mobile Multimedia Networks, 1998.

[2] L.Hsu, R. Purnadi and S.S. Peter Wang, Maintaining Quality of Service (QoS) during Handoff in Cellular System with Movement Prediction Schemes, IEEE 1999.

[3] Sunghyun Choi and Kang G. Shin, Predictive and Adaptive Bandwith Reservation for Han-doffs in QoS-Sensitive Cellular Networks, IEEE 1998

[4] X. Shen, J. W. Mark and J. Ye, User Mobility Profile Prediction: An Adaptive Fuzzy Interference Approach, Wireless Network 6, (2000) PP 362-374.

[5] D. Ashrook and T. Staruer, Learning Significant Locations and Predicting user Movement with GPS, Proceedings of the 6th International Symposium on Wearable Computers TSWC'02.

[6] W.S. Soh and Hyong S. Kim, QoS Provisioning in Cellular Networks Based on Mobility Prediction Techniques, IEEE Communication Magazine, January 2003, PP 86-92.

[7] S.C. Liou and H.C. Lu, Applied Neural Network for Location Prediction and Resource Reservation Scheme in Wireless Network, Proceedings of ICCT, 2003.

[8] J. Capka and R. Boutaba, Mobility Prediction in Wireless Networks using Neural Networks, International Federation for Information Proceeding, IFIP2004.

[9] N. Samaan, and A. Karmouch, A Mobility Perdiction Architecture based on Contextual Knowledge and Conceptual Maps, IEEE transactions on mobile computing, Vol4, No 6 November/December 2005

[10] J. M. François and G. Leduc, Entropy-Based Knowledge Spreading and Application to Mobility Prediction, ACM CoNEXT'05 October 24-27, 2005, Toulouse, France.

[11] M. Daoui, A. M'zoughi, M. Lalam, M. Belkadi and R. Aoudjit, Mobility prediction based on an ant system, Computer Communications, 31 (2008) 3090–3097.

[12] M. Daoui, A. M'zoughi, M. Lalam, R. Aoudjit and M. Belkadi, Forecasting models, methods and applications, mobility prediction in cellular network, i-concepts press, 2010 (221-232)

[13] L. Chamek, M. Daoui, M. Lalam. Mobility prediction based on classification according to the profile Journées sur les rencontres en Informatique R2I 12- 14 juin 2011

[14] M. Belkadi , R. Aoudjit, M. Daoui, M. Lalam Mobile Localization Based on Clustering, I. J. Computer Network and Information Security, 201 3, 9 , 37 -44

[15] P. Hu and J. Young, 1990 Nationwide Personal Transportation Survey (NPTS), Office of Highway Information Management, October 1994.

[16] J. Han, M. Kamber, Data mining concepts & techniques, Morgan Kaufmann, 2 edition, 2006

[17] J. Scourias and T. Kunz, An Activity-based Mobility Model and Location Management Simulation Framework, Proc., Second ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), 1999, PP. 61-68.