

International Journal on

Advances in Networks and Services



2015 vol. 8 nr. 1&2

The *International Journal on Advances in Networks and Services* is published by IARIA.

ISSN: 1942-2644

journals site: <http://www.ariajournals.org>

contact: petre@aria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Networks and Services, issn 1942-2644
vol. 8, no. 1 & 2, year 2015, http://www.ariajournals.org/networks_and_services/

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Networks and Services, issn 1942-2644
vol. 8, no. 1 & 2, year 2015, <start page>:<end page>, http://www.ariajournals.org/networks_and_services/

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA

www.aria.org

Copyright © 2015 IARIA

Editor-in-Chief

Tibor Gyires, Illinois State University, USA

Editorial Advisory Board

Mario Freire, University of Beira Interior, Portugal

Jens Martin Hovem, Norwegian University of Science and Technology, Norway

Vitaly Klyuev, University of Aizu, Japan

Noel Crespi, Institut TELECOM SudParis-Evry, France

Editorial Board

Ryma Abassi, Higher Institute of Communication Studies of Tunis (Iset'Com) / Digital Security Unit, Tunisia

Majid Bayani Abbasy, Universidad Nacional de Costa Rica, Costa Rica

Jemal Abawajy, Deakin University, Australia

Javier M. Aguiar Pérez, Universidad de Valladolid, Spain

Rui L. Aguiar, Universidade de Aveiro, Portugal

Ali H. Al-Bayati, De Montfort Uni. (DMU), UK

Giuseppe Amato, Consiglio Nazionale delle Ricerche, Istituto di Scienza e Tecnologie dell'Informazione (CNR-ISTI), Italy

Mario Anzures-García, Benemérita Universidad Autónoma de Puebla, México]

Pedro Andrés Aranda Gutiérrez, Telefónica I+D - Madrid, Spain

Cristian Anghel, University Politehnica of Bucharest, Romania

Miguel Ardid, Universitat Politècnica de València, Spain

Valentina Baljak, National Institute of Informatics & University of Tokyo, Japan

Alvaro Barradas, University of Algarve, Portugal

Mostafa Bassiouni, University of Central Florida, USA

Michael Bauer, The University of Western Ontario, Canada

Carlos Becker Westphall, Federal University of Santa Catarina, Brazil

Zdenek Becvar, Czech Technical University in Prague, Czech Republic

Francisco J. Bellido Outeiriño, University of Cordoba, Spain

Djamel Benferhat, University Of South Brittany, France

Jalel Ben-Othman, Université de Paris 13, France

Mathilde Benveniste, En-aerion, USA

Luis Bernardo, Universidade Nova of Lisboa, Portugal

Alex Bikfalvi, Universidad Carlos III de Madrid, Spain

Thomas Michael Bohnert, Zurich University of Applied Sciences, Switzerland

Eugen Borgoci, University "Politehnica" of Bucharest (UPB), Romania

Fernando Boronat Seguí, Universidad Politecnica de Valencia, Spain

Christos Bouras, University of Patras, Greece

Mahmoud Brahim, University of Msila, Algeria
Marco Bruti, Telecom Italia Sparkle S.p.A., Italy
Dumitru Burdescu, University of Craiova, Romania
Diletta Romana Cacciagrano, University of Camerino, Italy
Maria-Dolores Cano, Universidad Politécnica de Cartagena, Spain
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Eduardo Cerqueira, Federal University of Para, Brazil
Bruno Chatras, Orange Labs, France
Marc Cheboldaeff, T-Systems International GmbH, Germany
Kong Cheng, Telcordia Research, USA
Dickson Chiu, Dickson Computer Systems, Hong Kong
Andrzej Chydzinski, Silesian University of Technology, Poland
Hugo Coll Ferri, Polytechnic University of Valencia, Spain
Noelia Correia, University of the Algarve, Portugal
Noël Crespi, Institut Telecom, Telecom SudParis, France
Paulo da Fonseca Pinto, Universidade Nova de Lisboa, Portugal
Philip Davies, Bournemouth and Poole College / Bournemouth University, UK
Carlton Davis, École Polytechnique de Montréal, Canada
Claudio de Castro Monteiro, Federal Institute of Education, Science and Technology of Tocantins, Brazil
João Henrique de Souza Pereira, University of São Paulo, Brazil
Javier Del Ser, Tecnia Research & Innovation, Spain
Behnam Dezfooli, Universiti Teknologi Malaysia (UTM), Malaysia
Daniela Dragomirescu, LAAS-CNRS, University of Toulouse, France
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Wan Du, Nanyang Technological University (NTU), Singapore
Matthias Ehmann, Universität Bayreuth, Germany
Wael M El-Medany, University Of Bahrain, Bahrain
Imad H. Elhajj, American University of Beirut, Lebanon
Gledson Elias, Federal University of Paraíba, Brazil
Joshua Ellul, University of Malta, Malta
Rainer Falk, Siemens AG - Corporate Technology, Germany
Károly Farkas, Budapest University of Technology and Economics, Hungary
Huei-Wen Ferng, National Taiwan University of Science and Technology - Taipei, Taiwan
Gianluigi Ferrari, University of Parma, Italy
Mário F. S. Ferreira, University of Aveiro, Portugal
Bruno Filipe Marques, Polytechnic Institute of Viseu, Portugal
Ulrich Flegel, HFT Stuttgart, Germany
Juan J. Flores, Universidad Michoacana, Mexico
Ingo Friese, Deutsche Telekom AG - Berlin, Germany
Sebastian Fudickar, University of Potsdam, Germany
Stefania Galizia, Innova S.p.A., Italy
Ivan Ganchev, University of Limerick, Ireland
Miguel Garcia, Universitat Politècnica de Valencia, Spain
Emiliano Garcia-Palacios, Queens University Belfast, UK
Marc Gilg, University of Haute-Alsace, France
Debasis Giri, Haldia Institute of Technology, India

Markus Goldstein, Kyushu University, Japan
Luis Gomes, Universidade Nova Lisboa, Portugal
Anahita Gouya, Solution Architect, France
Mohamed Graiet, Institut Supérieur d'Informatique et de Mathématique de Monastir, Tunisie
Christos Grecos, University of West of Scotland, UK
Vic Grout, Glyndwr University, UK
Yi Gu, Middle Tennessee State University, USA
Angela Guercio, Kent State University, USA
Xiang Gui, Massey University, New Zealand
Mina S. Guirguis, Texas State University - San Marcos, USA
Tibor Gyires, School of Information Technology, Illinois State University, USA
Keijo Haataja, University of Eastern Finland, Finland
Gerhard Hancke, Royal Holloway / University of London, UK
R. Hariprakash, Arulmigu Meenakshi Amman College of Engineering, Chennai, India
Go Hasegawa, Osaka University, Japan
Eva Hladká, CESNET & Masaryk University, Czech Republic
Hans-Joachim Hof, Munich University of Applied Sciences, Germany
Razib Iqbal, Amdocs, Canada
Abhaya Induruwa, Canterbury Christ Church University, UK
Muhammad Ismail, University of Waterloo, Canada
Vasanth Iyer, Florida International University, Miami, USA
Peter Janacik, Heinz Nixdorf Institute, University of Paderborn, Germany
Imad Jawhar, United Arab Emirates University, UAE
Aravind Kailas, University of North Carolina at Charlotte, USA
Mohamed Abd rabou Ahmed Kalil, Ilmenau University of Technology, Germany
Kyoung-Don Kang, State University of New York at Binghamton, USA
Sarfraz Khokhar, Cisco Systems Inc., USA
Vitaly Klyuev, University of Aizu, Japan
Jarkko Kneckt, Nokia Research Center, Finland
Dan Komosny, Brno University of Technology, Czech Republic
Ilker Korkmaz, Izmir University of Economics, Turkey
Tomas Koutny, University of West Bohemia, Czech Republic
Evangelos Kranakis, Carleton University - Ottawa, Canada
Lars Krueger, T-Systems International GmbH, Germany
Kae Hsiang Kwong, MIMOS Berhad, Malaysia
KP Lam, University of Keele, UK
Birger Lantow, University of Rostock, Germany
Hadi Larijani, Glasgow Caledonian Univ., UK
Annett Laube-Rosenpflanzler, Bern University of Applied Sciences, Switzerland
Gyu Myoung Lee, Institut Telecom, Telecom SudParis, France
Shiguo Lian, Orange Labs Beijing, China
Chiu-Kuo Liang, Chung Hua University, Hsinchu, Taiwan
Wei-Ming Lin, University of Texas at San Antonio, USA
David Lizcano, Universidad a Distancia de Madrid, Spain
Chengnian Long, Shanghai Jiao Tong University, China
Jonathan Loo, Middlesex University, UK

Pascal Lorenz, University of Haute Alsace, France
Albert A. Lysko, Council for Scientific and Industrial Research (CSIR), South Africa
Pavel Mach, Czech Technical University in Prague, Czech Republic
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Damien Magoni, University of Bordeaux, France
Ahmed Mahdy, Texas A&M University-Corpus Christi, USA
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Gianfranco Manes, University of Florence, Italy
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Moshe Timothy Masonta, Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa
Hamid Menouar, QU Wireless Innovations Center - Doha, Qatar
Guowang Miao, KTH, The Royal Institute of Technology, Sweden
Mohssen Mohammed, University of Cape Town, South Africa
Miklos Molnar, University Montpellier 2, France
Lorenzo Mossucca, Istituto Superiore Mario Boella, Italy
Jogesh K. Muppala, The Hong Kong University of Science and Technology, Hong Kong
Katsuhiko Naito, Mie University, Japan
Deok Hee Nam, Wilberforce University, USA
Sarmistha Neogy, Jadavpur University- Kolkata, India
Rui Neto Marinheiro, Instituto Universitário de Lisboa (ISCTE-IUL), Instituto de Telecomunicações, Portugal
David Newell, Bournemouth University - Bournemouth, UK
Armando Nolasco Pinto, Universidade de Aveiro / Instituto de Telecomunicações, Portugal
Jason R.C. Nurse, University of Oxford, UK
Kazuya Odagiri, Yamaguchi University, Japan
Máirtín O'Droma, University of Limerick, Ireland
Rainer Oechsle, University of Applied Science, Trier, Germany
Henning Olesen, Aalborg University Copenhagen, Denmark
Jose Oscar Fajardo, University of the Basque Country, Spain
Constantin Paleologu, University Politehnica of Bucharest, Romania
Eleni Patouni, National & Kapodistrian University of Athens, Greece
Harry Perros, NC State University, USA
Miodrag Potkonjak, University of California - Los Angeles, USA
Yusnita Rahayu, Universiti Malaysia Pahang (UMP), Malaysia
Yenumula B. Reddy, Grambling State University, USA
Oliviero Riganelli, University of Milano Bicocca, Italy
Teng Rui, National Institute of Information and Communication Technology, Japan
Antonio Ruiz Martinez, University of Murcia, Spain
George S. Oreku, TIRDO / North West University, Tanzania/ South Africa
Sattar B. Sadkhan, Chairman of IEEE IRAQ Section, Iraq
Husnain Saeed, National University of Sciences & Technology (NUST), Pakistan
Addisson Salazar, Universidad Politecnica de Valencia, Spain
Sébastien Salva, University of Auvergne, France
Ioakeim Samaras, Aristotle University of Thessaloniki, Greece
Luz A. Sánchez-Gálvez, Benemérita Universidad Autónoma de Puebla, México
Teerapat Sanguankotchakorn, Asian Institute of Technology, Thailand
José Santa, University Centre of Defence at the Spanish Air Force Academy, Spain

Rajarshi Sanyal, Belgacom International Carrier Services, Belgium
Mohamad Sayed Hassan, Orange Labs, France
Thomas C. Schmidt, HAW Hamburg, Germany
Hans Scholten, Pervasive Systems / University of Twente, The Netherlands
Véronique Sebastien, University of Reunion Island, France
Jean-Pierre Seifert, Technische Universität Berlin & Telekom Innovation Laboratories, Germany
Sandra Sendra Compte, Polytechnic University of Valencia, Spain
Dimitrios Serpanos, Univ. of Patras and ISI/RC ATHENA, Greece
Roman Y. Shtykh, Rakuten, Inc., Japan
Salman Ijaz Institute of Systems and Robotics, University of Algarve, Portugal
Adão Silva, University of Aveiro / Institute of Telecommunications, Portugal
Florian Skopik, AIT Austrian Institute of Technology, Austria
Karel Slavicek, Masaryk University, Czech Republic
Vahid Solouk, Urmia University of Technology, Iran
Peter Soreanu, ORT Braude College, Israel
Pedro Sousa, University of Minho, Portugal
Vladimir Stantchev, SRH University Berlin, Germany
Radu Stoleru, Texas A&M University - College Station, USA
Lars Strand, Nofas, Norway
Stefan Strauß, Austrian Academy of Sciences, Austria
Álvaro Suárez Sarmiento, University of Las Palmas de Gran Canaria, Spain
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Young-Joo Suh, POSTECH (Pohang University of Science and Technology), Korea
Junzhao Sun, University of Oulu, Finland
David R. Surma, Indiana University South Bend, USA
Yongning Tang, School of Information Technology, Illinois State University, USA
Yoshiaki Taniguchi, Kindai University, Japan
Anel Tanovic, BH Telecom d.d. Sarajevo, Bosnia and Herzegovina
Olivier Terzo, Istituto Superiore Mario Boella - Torino, Italy
Tzu-Chieh Tsai, National Chengchi University, Taiwan
Samyr Vale, Federal University of Maranhão - UFMA, Brazil
Dario Vieira, EFREI, France
Natalija Vlajic, York University - Toronto, Canada
Lukas Vojtech, Czech Technical University in Prague, Czech Republic
Michael von Riegen, University of Hamburg, Germany
You-Chiun Wang, National Sun Yat-Sen University, Taiwan
Gary R. Weckman, Ohio University, USA
Chih-Yu Wen, National Chung Hsing University, Taichung, Taiwan
Michelle Wetterwald, HeNetBot, France
Feng Xia, Dalian University of Technology, China
Kaiping Xue, USTC - Hefei, China
Mark Yampolskiy, Vanderbilt University, USA
Dongfang Yang, National Research Council, Canada
Qimin Yang, Harvey Mudd College, USA
Beytullah Yildiz, TOBB Economics and Technology University, Turkey
Anastasiya Yurchyshyna, University of Geneva, Switzerland

Sergey Y. Yurish, IFSA, Spain

Jelena Zdravkovic, Stockholm University, Sweden

Yuanyuan Zeng, Wuhan University, China

Weiliang Zhao, Macquarie University, Australia

Wenbing Zhao, Cleveland State University, USA

Zibin Zheng, The Chinese University of Hong Kong, China

Yongxin Zhu, Shanghai Jiao Tong University, China

Zuqing Zhu, University of Science and Technology of China, China

Martin Zimmermann, University of Applied Sciences Offenburg, Germany

CONTENTS

pages: 1 - 16

Application-Oriented On-Demand Data Collection in Sparse Underwater Acoustic Sensor Networks Using Mobile Elements

Jalaja Janardanan Kartha, National Institute of Technology, Calicut, India
Lillykutty Jacob, National Institute of Technology, Calicut, India

pages: 17 - 26

The Challenge of On-demand Routing Protocols Improvement in Mobility Context

Tiguiane Yélékou, Polytechnic University of Bobo-Dioulasso, Burkina Faso
Philippe Meseure, University of Poitiers, XLIM-SIC CNRS Lab, France
Anne-Marie Poussard, University of Poitiers, XLIM-SIC CNRS Lab, France
Toundé Mesmin Dandjinou, Polytechnic University of Bobo-Dioulasso, Burkina Faso

pages: 27 - 41

HQMR: Hybrid QoS based Routing Protocol for Wireless Mesh Environment

Hajer Bargaoui, LE2I Laboratory, University of Burgundy, France
Nader Mbarek, LE2I Laboratory, University of Burgundy, France
Olivier Togni, LE2I Laboratory, University of Burgundy, France
Mounir Frikha, MEDIATRON Laboratory, SUPCOM, Tunisia

pages: 42 - 53

On the Performance of a Flow Aggregation Scheme for Seamless QoS and Utility Oriented Mobility Support in Wireless Mesh Networks

Salvatore Vanini, SUPSI, Switzerland
Dario Gallucci, SUPSI, Switzerland
Silvia Giordano, SUPSI, Switzerland
Maciej Urbanski, Poznan University of Technology, Poland
Przemyslaw Walkowiak, Poznan University of Technology, Poland

pages: 54 - 68

Evaluation of an Uncertainty Aware Hybrid Clock Synchronisation System for Wireless Sensor Networks

Christoph Steup, Otto-von-Guericke University, Germany
Sebastian Zug, Otto-von-Guericke University, Germany
Jörg Kaiser, Otto-von-Guericke University, Germany

pages: 69 - 80

Robust Timing Synchronization Preamble for MIMO-OFDM Systems Using Mapped CAZAC Sequences

Ali Rachini, INSA de Rennes, France
Fabienne Nouvel, INSA de Rennes, France
Bilal Beydoun, GET/UL - Lebanese University, Lebanon
Ali Beydoun, GET/UL - Lebanese University, Lebanon

pages: 81 - 91

Context-, Resource-, and User-Aware Provision of Services on Mobile Devices

André Ebert, Institute for Informatics, Mobile and Distributed Systems Group, LMU Munich, Germany

Florian Dorfmeister, Institute for Informatics, Mobile and Distributed Systems Group, LMU Munich, Germany
Marco Maier, Institute for Informatics, Mobile and Distributed Systems Group, LMU Munich, Germany
Claudia Linnhoff-Popien, Institute for Informatics, Mobile and Distributed Systems Group, LMU Munich, Germany

pages: 92 - 105

Server and Path Selection in a Light Architecture Content Streaming System with Dual Adaptation

Eugen Borcoci, University POLITEHNICA of Bucharest, Romania
Marius Vochin, University POLITEHNICA of Bucharest, Romania
Mihai Constantinescu, University POLITEHNICA of Bucharest, Romania
Jordi Mongay Batalla, Warsaw University of Technology / National Institute of Telecommunications Warsaw, Poland
Daniel Negru, LaBRI Lab, University of Bordeaux, France

pages: 106 - 117

User Preferences and Segments in App Store Marketing: A Conjoint-based Approach

Stephan Böhm, RheinMain University of Applied Sciences, Germany
Stefan Schreiber, RheinMain University of Applied Sciences, Germany
Wendy Colleen Farrell, RheinMain University of Applied Sciences, Germany

pages: 118 - 129

A Network-disaster Recovery System using Multiple-backup Operation Planes

Toshiaki Suzuki, Hitachi, Ltd., Japan
Hideki Endo, Hitachi, Ltd., Japan
Isao Shimokawa, Hitachi, Ltd., Japan
Kenichi Sakamoto, Hitachi, Ltd., Japan
Hidenori Inouchi, Hitachi, Ltd., Japan
Taro Ogawa, Hitachi, Ltd., Japan
Takanori Kato, Hitachi, Ltd., Japan
Akihiko Takase, Hitachi, Ltd., Japan

Application-Oriented On-Demand Data Collection in Sparse Underwater Acoustic Sensor Networks Using Mobile Elements

Jalaja Janardanan Kartha and Lillykutty Jacob

Department of Electronics and Communication Engineering
National Institute of Technology
Calicut, India

Emails: {jalaja@rit.ac.in, lilly@nitc.ac.in}

Abstract—Underwater Wireless Sensor Network (UWSN) is a group of sensors and underwater vehicles, networked via acoustic links, which performs collaborative tasks and enables a wide range of aquatic applications. Due to hostile environment, resource constraints and peculiarities of the underlying physical layer technology, providing energy-efficient data collection in a sparse UWSN is a challenging problem. We consider mobility-assisted routing technique for enabling connectivity and improving the energy efficiency of sparse UWSN, considering it as a Delay/Disruption Tolerant Network (DTN) or Intermittently Connected Network (ICN). We use analytical models to investigate the performance of the data collection scheme. Based on the result that the DTN scheme improves energy efficiency and Packet Delivery Ratio (PDR) at the cost of increased message latency, we investigate techniques to improve its delay performance. The effects of using multiple mobile elements for data collection and activity-based priority-polling are investigated. In addition, the suitability of a hybrid architecture and hierarchical organization of mobile elements for supporting delay-sensitive applications in the mobility-assisted framework, is explored. The analytical results are validated through extensive simulations using NS-2 based Aqua-Sim simulator. The results show that our model for on-demand data collection can effectively capture the underwater acoustic network conditions and facilitate performance evaluation of event-driven data collection in sparse UWSNs prior to deployment. The improved DTN framework shows superior performance in terms of energy efficiency and successful data delivery over ad-hoc multi hop network, and in terms of message latency, fairness and buffer space requirement over simple polling-based DTN framework.

Keywords—Underwater Sensor Networks; Delay Tolerant Network; Mobile Collector; Polling; Exhaustive Service; Fairness; Energy Efficiency; Hybrid Architecture.

I. INTRODUCTION

This paper extends our earlier work [1] presented at the Tenth International Conference on Wireless and Mobile Communication (ICWMC 2014), proposing two strategies for supporting delay-sensitive applications in mobility-assisted underwater data collection. Compared to the original paper, it contains more detailed analysis of the system model and introduces additional proposals for improving latency performance according to application requirements.

Underwater Wireless Sensor Networks (UWSNs) have emerged as powerful systems for providing autonomous support for several activities like oceanographic data collection, marine surveillance, disaster prediction, assisted navigation etc. As illustrated in Fig. 1, UWSN consists of a

number of different types of sensor nodes and autonomous underwater vehicles (AUVs) used for collaborative monitoring tasks. Ordinary underwater (UW) sensor nodes deployed at different depths are used to sense the environment and generate data. UW sink nodes are responsible for collecting this data and forwarding it to the surface sink. Surface sinks are equipped with RF communication link with the on-shore control centre and other surface sinks, acoustic links with the underwater sensor nodes, and an optional fibre optic link with the UW sink. Since electromagnetic waves are heavily attenuated in the salty sea water and optical signals are affected by scattering, acoustic communication is the underlying physical layer technology used in UWSNs. Development of underwater acoustic communication systems for interesting practical applications are available in [2], [3], [4] and [5]. Features like high latency, low bandwidth, high error probability and 3-dimensional deployment make the UWSNs significantly different from terrestrial WSNs [6].

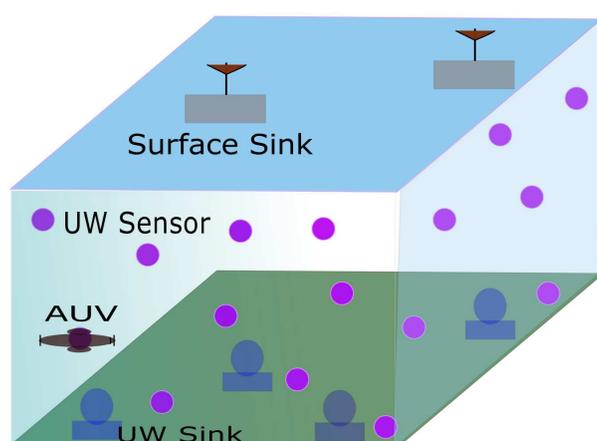


Figure 1. Underwater Sensor Network

The energy saving/efficiency is a critical issue for UWSN because of the high cost of deploying and/or re-deploying underwater equipment. Underwater sensors are expensive, mainly because of their more complex transceivers, and the ocean area that needs to be sensed is quite large.

Hence, UWSN deployment can be much sparser compared with terrestrial WSNs. Due to sparse deployment, harsh environment, node mobility and resource limitations, the network can be easily partitioned and a contemporaneous path may not exist between any two nodes. This results in sparse UWSNs that need to be treated as Intermittently Connected Networks (ICN) or Delay / Disruption Tolerant Networks (DTN) [7]. At any given time, when no path exists between source and destination, network partition is said to occur. DTNs are characterised by frequent partitions and potentially long message delivery delays. Such networks may never have an end-to-end contemporaneous path and traditional routing protocols are not practical since packets will be dropped when no routes are available.

The primary objective of a DTN routing protocol is to obtain high message delivery ratios with satisfactory latency performance, while maintaining low overhead. The characteristics of DTN are quite different from that of Internet and hence new system architectures and routing protocols are required for DTNs. DTN routing protocols can be generally classified as *forwarding-based* and *replication-based*. *Forwarding-based* schemes keep one copy of a message in the network and tries to forward that copy towards the sink at the earliest possible forwarding opportunity. *Replication-based* approaches like multipath routing are resource-hungry and hence not suitable for resource-constrained underwater applications. *Forwarding-based* approaches are limited in their effectiveness due to instability (or even non-existence) of routes from any particular node to the destination. To combat intermittent connectivity in resource-constrained UWSNs, a natural solution is to extend the store-and-forward routing to store-carry-and-forward routing. Proactive mobility of special mobile nodes can be made use of, to improve message delivery ratio and to reduce energy consumption. Since the next hop may not be immediately available for the current node to forward data, it has to buffer the data until it detects a *contact* or forwarding opportunity.

The three main approaches reported in the literature for data collection in wireless sensor networks, in general, are [8]: (i) Base Station (BS) approach, which uses direct communication between the source and the sink; (ii) Ad hoc network, which uses a multi-hop path from the source to the sink; and (iii) Mobility assisted routing, which makes use of a mobile sink or mobile relays for data collection. The first approach provides fast delivery, but suffers from reduced life time of sensors due to increased requirement of communication energy when the source to sink distance is large. The ad hoc multi hop network provides medium delay with medium power requirement, but suffers from the 'hot spot' problem or the sink neighbourhood problem. In addition, an end-to-end contemporaneous path should exist for successful data collection, which is not always possible in the harsh marine environment in which UWSNs are deployed.

Mobility assisted routing approach supports the DTN concept of store-carry-and-forward and fills connectivity gaps in the network. Also, it reduces transmit power consumption and eliminates the relaying overhead. However, due to the limited travel speed of the mobile elements, data collection latency will be large, but such large latency may be acceptable in certain environmental sensing applications which are not time-critical. Typical example of such an

application is the continuous monitoring and recording of the behaviour of underwater plates in tectonics, for later scientific analysis. Compared with periodic data collection in which the locations are given and fixed, event-driven data collection can shorten the response time, and can thereby support delay-sensitive applications. The arrival of events that require attention need not be deterministic and planned; instead, they can be online and stochastic. However, there exists no proper model for analyzing the performance of mobility-assisted event-driven data collection scheme in UWSNs. Investigation of event-driven on-demand data collection using energy-efficient mobility-assisted scheme in sparse UWSNs and enhancing it for supporting delay-sensitive applications like pollution monitoring and earthquake prediction, is the focus of this paper.

We start with a basic DTN framework for energy efficient data collection in sparse UWSNs using a single mobile sink; and then augment it with techniques to improve its data collection performance by: (i) introducing priority; (ii) employing multiple data collectors; (iii) deploying a hybrid architecture with both static and mobile sinks; and (iv) organizing the mobile collectors in a hierarchical structure. Analytical results for energy efficiency, packet delivery ratio (PDR), message latency, and sensor buffer occupancy are presented. The analytical results are validated using our own simulation model developed in Aqua-Sim [9], an NS-2 [10] based network simulator, developed by the University of Connecticut. The major contributions of this paper include: (i) Investigating an energy-efficient DTN framework for event-driven data collection in sparse UWSNs; (ii) Analyzing the performance metrics of the proposed framework; (iii) Proposing techniques to enhance the latency performance; and (iii) Developing the simulation model for validation of analytical results and further research. The rest of the paper is organized as follows. A brief review of the related work is given in Section II. The basic system model is presented in Section III and the expressions used for analytical results with this model are developed in Section IV. Techniques for delay performance enhancement of the basic model are discussed and analyzed in Section V. Section VI discusses the analytical and simulation results. The paper is concluded in Section VII.

II. RELATED WORK

Several routing protocols have been developed for UWSNs, most of them suitable only for connected networks. A detailed review of different routing techniques for UWSNs is given in [11] and a comparative analysis of routing protocols is done in [12]. Vector Based Forwarding (VBF) [13] is a typical geographical routing protocol and Hop-by-hop Vector-based forwarding (HH-VBF) [14] is its more energy-efficient version, better suited for sparse networks. Both VBF and HH-VBF do not support mobility-assisted data collection and they require the network to be connected. Energy analysis of routing protocols for UWSNs is presented by Domingo [15] and by Zorzi et al. [16]. An approach for minimization of energy consumption in multi-hop UWSNs is proposed by Geethu et al. in [17]. Javaid et al. have proposed delay-sensitive routing schemes for UWSNs in [18] and chain based communication in cylindrical UWSNs in [19].

Recently, considerable effort has been devoted to developing architectures and routing algorithms for terrestrial

DTNs. Routing in DTNs is investigated by Jain et al. [20] and underwater DTN routing is discussed by Tolba et al. [21]. Adaptive data collection in sparse UWSNs using mobile elements is proposed by the authors in [22]. A message ferrying approach for data delivery in sparse mobile ad hoc networks is presented in [23]. Guo et al. have proposed an adaptive routing protocol for UWSNs, considering it as a DTN [24]. Shah et al. [8] have presented a three-tier architecture based on mobility to address the problem of energy efficient data collection in a terrestrial sensor network. The same architecture with an enhanced analytical model has been presented by Jain et al. [25]. An M/G/1 queueing model is used by He et al. [26] for mobility-assisted routing, proposed for reducing and balancing the energy consumption of sensor nodes. The use of controlled mobility for low energy embedded networks has been discussed by Arun et al. [27]. AUV-aided routing for UWSNs is discussed by Yoon et al. [28] and Hollinger et al. [29]. A mobile geocast routing protocol for efficient data collection from underwater sensor nodes is proposed by Chen et al. in [30]. Polling-based scheduling in body sensor networks has been discussed by Motoyama [31] and the usage of message ferries in ad hoc networks is considered by Kavitha et al. [32]. Delay and lifetime performance of mobility-assisted periodic data collection in sparse UWSNs is presented by the authors in [33].

Even though the development of routing protocols for dense/connected UWSNs and the adaptation of DTN approaches for terrestrial sensor networks has already been addressed thoroughly, the energy-efficient data collection in resource-constrained sparse UWSNs has not been adequately investigated. In addition, proper analytical models and simulation environment for the evaluation of all performance metrics and for the study of trade-offs in different data collection schemes are still lacking. Since field tests in the ocean bottom are costly and mostly infeasible prior to sensor deployment, realistic models will be extremely useful for designing application-oriented networks. Also, the reported DTN schemes in UWSNs are either resource-hungry or not suitable for on-demand data collection applications. Most of the mobility-assisted data collection schemes for sensor networks focus on the offline scenario, where the data collection is carried out in a periodic manner. A potential problem with this periodic data collection is that, certain sensor nodes may not have data to upload, and visiting them just to find that no data to collect is not efficient.

The adaptation of mobility-assisted schemes for event-driven on-demand data collection in UWSNs and the enhancement of DTN schemes for delay-sensitive applications are still unexplored. In this paper, we first propose an energy-efficient on-demand data collection scheme suitable for non-time-critical applications in UWSNs and then we augment it with techniques to support delay-sensitive applications.

III. SYSTEM MODEL

We consider large and sparse UWSNs with possibly disconnected components and with mobile elements used for data collection. Though both 3-dimensional and 2-dimensional deployments are possible, we limit our study to 2-dimensional network as shown in Fig. 2, with sensor nodes anchored to the ocean bottom. The static sensors monitor the underwater

surroundings, generate data and store it in the sensor buffer. They have limited non-rechargeable battery power and they communicate using acoustic links. The underwater sink, acting like a base station (BS) is responsible for gathering the sensed data from the static sensors by employing mobile collectors (MCs) and forwarding the collected data to the surface sink. MCs are mobile entities with large processing and storage capacity, renewable power, and the ability to communicate with static sensors, underwater sink and other MCs (if any).

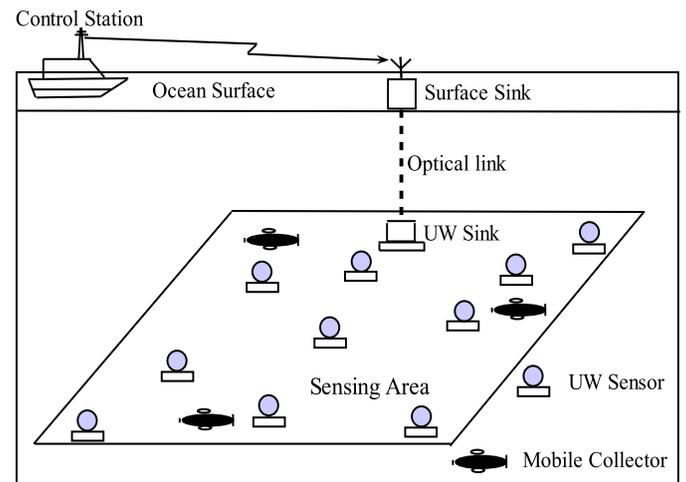


Figure 2. System Model : 2-D network with static sensors and MCs

When an event of interest occurs, the static sensors initiate data collection requests to the BS using direct or ad hoc multi-hop communication. The service request packet is assumed to be very short compared to data packets and the former will contain location information of the node, sensor buffer occupancy, priority of application, and any other relevant information like packet arrival rate or the delay-sensitivity of request. The BS will collect the requests and based on the system load and the delay constraints, it can decide the number of MCs needed and the sequence of visiting the nodes by each MC. Accordingly, BS will create one or more visit tables specifying the order of visiting the nodes and schedule the required number of MCs with a unique visit table assigned to each one of it. It maintains a service queue for the received data collection requests, and serve them with the first-come-first serve (FCFS) discipline. Serving a request means, the MC moves into the proximity of the corresponding sensor node and collects data from it.

As an MC moves in close proximity to (i.e., within transmission range of) a static sensor, the sensor's data is transferred to the MC and buffered there for further processing. We consider proactive controlled mobility of the MC, as the random mobility will fail to give latency bounds. Each MC will visit the sensor, collect the buffered data and proceed towards the next node in the visit table and this process is repeated. Sensors' bulk data communications are limited to transferring data to a nearby MC, so as to reduce energy consumption. Since the MCs are assumed to be resource-unconstrained and the BS (i.e., UW sink) is assumed to have a high speed link with the surface sink, we restrict our study to the collection of data from the static sensors deployed at the ocean bottom by the MC(s) travelling with a constant speed and pausing at the

vicinity of sensors for data collection. The data is assumed to have been successfully delivered once it has been collected by the MCs.

IV. ANALYTICAL STUDY

In this section, we develop the necessary analytical expressions, the numerical results of which are compared with the simulation results in Section VI. All the features of acoustic channel, propagation and devices significantly affect the performance measures of the UWSN and hence the performance of data collection schemes.

A. Energy Efficiency

One important motivation for employing a mobile sink is that it increases the lifetime of the network by balancing the energy consumption of the sensor nodes. The energy consumption of the static nodes alone is considered, since the mobile node is assumed to be rechargeable or having much higher initial energy compared to the static sensors. The energy consumed by the static sensor nodes for sensing and processing is negligible compared with that for underwater acoustic data transmission, and hence we consider the energy consumption for data transmission only.

Underwater Channel: Propagation of sound underwater is at a very low speed of 1500 m/s and it occurs over multiple paths. Underwater acoustic communication channels are characterized by a path loss that depends not only on the distance between the source and the sink, but also on the signal frequency. Path loss is the sum of absorption loss (due to the transfer of acoustic energy in to heat) and spreading loss (due to the regular weakening of a sound signal as it moves outwards from the source). At shorter ranges, spreading loss plays a proportionally larger role compared with absorption loss. Spreading loss is frequency-independent, but depends on the geometry, where as the absorption loss increases with frequency.

The SNR of an emitted underwater signal at the receiver is expressed by the passive sonar equation as [34]

$$SNR = SL - TL - NL + DI \quad (1)$$

where SL is the source level, TL is the transmission loss, NL is the noise level, and DI is the directivity index. A micro pascal (μPa) is a measurement of pressure commonly applied to underwater sound. All quantities in (1) are in dB re μPa where the reference value 1 μPa corresponds to the intensity value of $0.67 \times 10^{-18} \text{ W/m}^2$. Assuming a target SNR of 20 dB at the receiver, an ambient noise level of 70 dB (which is representative of underwater environments), and omnidirectional antennas for transmission and reception, we have the required source level $SL = TL + 90 \text{ dB}$.

The transmission loss or the attenuation factor $A(l, f)$ of an underwater acoustic channel for a distance l and frequency f is given by (2) as [34]:

$$10 \log A(l, f) = k \cdot 10 \log l + l \cdot 10 \log a(f) \quad (2)$$

where the first term is the spreading loss and the second term is the absorption loss. The spreading coefficient $k = 1$ for cylindrical spreading (shallow water scenario) and $k = 2$ for

spherical case (deep water scenario). Thorp's formula [34] is used to express the absorption coefficient as:

$$10 \log a(f) = \frac{0.11f^2}{1 + f^2} + \frac{44f^2}{4100 + f^2} + \frac{2.75f^2}{10^4} + 0.003 \quad (3)$$

The dependence of absorption loss on signal frequency implies the dependence of available acoustic bandwidth on communication distance. The resulting bandwidth limitation is a fundamental one due to the physics of acoustic propagation. Typical bandwidth of underwater channel is of the order of a few kilohertz at 10-100 km and 10 kilohertz at 1-10 km.

For a given target signal-to-noise ratio SNR_{tgt} at receiver, available bandwidth $B(l)$, and noise power spectral density $N(f)$, the required transmit power $P_t(l)$ can be expressed as a function of the transmitter-receiver distance l [16]. If P_r is the receive power, L is the packet size in bits, and α is the bandwidth efficiency of modulation, the energy consumption for the single hop data transfer of one packet becomes

$$E_{hop}(l) = \frac{P_r + P_t^{el}(l)L}{\alpha B(l)} \quad (4)$$

where $P_t^{el}(l)$ is the electrical power (in watts) corresponding to $P_t(l)$ in dB re μPa . Compared to P_r , P_t^{el} is very large and hence its contribution to the energy consumption of sensor nodes is significant. It is clear that the power consumption for data transfer over a single hop of length l increases with l , while the available bandwidth decreases with l . Hence, short range high bandwidth communication is to be adopted, whenever possible, to minimize energy consumption.

In order to investigate the superiority of the MC-based DTN model in conserving energy, let us compare the energy overhead associated with transferring one packet from the sensor to the BS using the ad hoc multi-hop approach and the *store-carry-and-forward* DTN approach. For this analysis purpose, we assume the network to be well connected so that ad hoc multi hop communication is possible from each sensor to the BS located at the centre. Also, for tractability of analysis, without losing generality, we assume the network layout to be circular. In order to quantify the potential savings in energy, we follow an approach similar to that of [27] with and without using a mobile node.

Assume N static sensor nodes with transmission range r are randomly and uniformly deployed over a circular area A of radius R with the sink located at the centre. We can calculate the minimum energy requirement of a node for transferring one packet generated by each node to the sink, using ad hoc multi-hop approach. Assuming ideal MAC such that no collisions occur, the packets originated by the nodes within a distance r from the sink need to be sent to the sink directly, whereas those generated by nodes at larger distances need to be relayed by the inner nodes towards the sink.

If every static node located in the k th annulus of the circular area generates one packet, then the minimum number of transmissions due to packets originated from the k th annulus is $MinTx(k) = N \frac{A(k)}{A} k$, where $A(k)$ is the area of the k^{th} annulus and $k = 1$ for the innermost annulus. In the mobility-assisted data collection, irrespective of the position of the nodes, each static node transmits only the packets generated by it. Instead, in the case of multi-hop architecture, if every

node generates 1 packet each, for a large value of N , on an average, the number of receptions and transmissions to be undertaken by a node in annulus k will be, respectively, $NodeRx(k) = \frac{A(k+1)}{A(k)}NodeTx(k+1)$ and $NodeTx(k) = 1 + \frac{A(k+1)}{A(k)}NodeTx(k+1)$, except for the outermost annulus ($k = \lceil \frac{R}{r} \rceil$) where the corresponding values are 0 and 1.

The above analysis shows the increased relaying overhead of a sensor node with its proximity to the sink. If we define the *Energy Overhead Factor* (EOF) of a node as the ratio of the total number of transmissions from the node to the number of transmissions corresponding to the packets originated at that node, it is seen that all the sensor nodes have the same EOF (equal to 1 with an error-free channel) in MC-based scheme, while it is approximately equal to $NodeTx(k)$ in multihop network. High *Energy Overhead Factor* implies low energy efficiency.

A natural consequence of this unbalanced usage of stored battery power by the static sensor nodes in the adhoc multi hop network is the non uniformity in the residual energy of the sensor nodes after operation for a fixed amount of time. If E_i is the initial energy of a node, the maximum number of packet transmissions over a hop distance l that can be afforded by it before being completely drained off its energy will be $\frac{E_i}{E_{hop}(l)}$. Due to the absence of relaying overhead in MC-based architecture, the residual energy of all the nodes will be uniformly distributed in the network. At the same time, in the ad hoc multi hop network, due to the increased relaying overhead of the sensors with proximity to sink, residual energy of the 1-hop neighbours of the sink will be considerably small, compared to that of the nodes along the periphery of the network. This sink neighbourhood problem leads to premature death of the 1-hop neighbours of the sink, thus resulting in the disconnection of the sink from the rest of the network which means that the usefulness of the network is lost. Hence, in applications in which network lifetime is more important than message delay, mobility-assisted routing is the best option. In addition, in disconnected or partitioned networks in which both direct and ad-hoc multi hop communications are too costly in terms of energy consumption, the proposed mobility-assisted framework is the only option.

B. Data Collection Latency

Due to the mechanical movement of the MC to provide connectivity and facilitate data collection from the sensors located quite far apart, the latency of the sensed data in the mobility-assisted approach will be much larger compared to that in the other two approaches. In addition, since the sensed data is to be buffered till the next visit of the MC, buffer overflow and packet loss will occur if the sensor buffer space is not sufficient or if the inter-arrival time of the MC at a sensor is too high. Realistic estimation of these parameters using a proper analytical model is essential to assess the suitability of the proposed scheme for a particular application, based on the requirements of the application and the resource constraints of the network. A model matching the mobility-assisted on-demand data collection framework is a multiple-queue single-server queueing model or a *polling*

model; a system of multiple queues accessed in cyclic or other specified sequence by a single server.

The traditional polling system consists of N infinite size queues and a single server that serves them one at a time [35]. The arrival process to queue i is assumed to be an independent Poisson process with rate λ_i . The customers arriving to queue i are assumed to have service time X_i , which is a random variable with first and second moments $E[X_i]$ and $E[X_i^2]$, respectively. After being served at queue i , the customer is assumed to leave the system. In the basic polling model, the server visits (or polls) the queues in a cyclic order and after completing a visit to queue i , it incurs a switch over period or *walk time*. The period during which the server continuously serves queue i is called the *service period* of queue i and the preceding period is called the *switch over* period of queue i . Mobile Collector and the static sensor buffers in our model correspond to the single server and queues of the polling model, respectively. Packets buffered in the sensor buffer, waiting for a transmission opportunity, are analogous to the customers waiting for service in the polling model. Travel time of the MC to move from one location to the next is modelled as the *walk time* and the sojourn time at each location to transfer data from the near by sensor's buffer to the MC is modelled as the *service time*.

According to the instant at which the MC leaves the sensor, different service policies are available, which prescribes how the packets (if any) from each sensor buffer will be collected. The important service policies applicable to on-demand data collection are: *Exhaustive*, *Gated* and *1-Limited*. In the *1-Limited* policy, at most one packet is collected from a sensor buffer at each visit. In the *Exhaustive* service, upon visiting a sensor, the MC collects all the packets until no more packets are available at that sensor buffer. In *gated* service, MC collects only those packets which are queued at its arrival instant. In other words, the packets that arrive during the course of the current data collection operation are not considered, where as under *Exhaustive* service policy, the MC collects the packets buffered so far plus the packets being generated when the already buffered packets are being transferred.

The inter-arrival times of service requests are assumed to be independent of MC travel time and data collection time. Assuming Poisson arrival of packets at rate λ_i at sensor buffer i , the traffic load at sensor i is defined by $\rho_i = \lambda_i E[X_i]$, $1 \leq i \leq N$, and the total offered load in the system is given by $\rho = \sum_{i=1}^N \rho_i$, where $E[X_i]$ is the mean packet transfer time. For system stability, ρ should be less than 1. If the mean of the total walk time is denoted by R , the mean cycle time of the MC is given by [35]

$$E[C] = \frac{R}{1 - \rho} \quad (5)$$

For system stability, all packets that arrive during a cycle of the MC must be served during a cycle time. Hence, the mean service period for sensor buffer i during a cycle time will be

$$E[S_i] = E[X_i]\lambda_i E[C] = \frac{\rho_i R}{1 - \rho} \quad (6)$$

and the mean number of packets collected from sensor buffer i in a polling cycle will be

$$E[\Phi_i] = \lambda_i E[C] \quad (7)$$

Using (5) and (6), the average inter-arrival time of the MC at a sensor buffer can be evaluated as

$$E[I_i] = E[C] - E[S_i] = \frac{(1 - \rho_i)R}{1 - \rho} \quad (8)$$

We assume that the stability condition is achieved and the system is in steady state. The main performance measure in data collection is the mean waiting time of a packet in the sensor buffer, the exact analysis of which is difficult. Hence, we resort to the *pseudo-conservation law* based on the *stochastic decomposition* of unfinished work in an infinite-buffer polling system [35]. For analytical tractability, we assume a symmetric system with equal data generation rate λ and equal mean packet service time X at all sensors. Let the MC travel time between two consecutive locations be a random variable with mean and variance $E[Y]$ and Δ^2 , respectively. Under the assumption of *exhaustive* service, The mean waiting time of the packet in the sensor buffer before the MC approaches it for data transfer can be obtained as [35]:

$$E[WQ]_{exh} = \frac{\Delta^2}{2E[Y]} + \frac{N\lambda E[X^2] + E[Y](N - \rho)}{2(1 - \rho)} \quad (9)$$

With *gated* and *1-Limited* service policies, the expressions for mean waiting time become

$$E[WQ]_{gated} = \frac{\Delta^2}{2E[Y]} + \frac{N\lambda E[X^2] + E[Y](N + \rho)}{2(1 - \rho)} \quad (10)$$

$$E[WQ]_{lim} = \frac{\Delta^2}{2E[Y]} + \frac{N\lambda E[X^2] + E[Y](N + \rho) + N\lambda\Delta^2}{2(1 - \rho - N\lambda E[Y])} \quad (11)$$

At light loads (ρ approaching 0), the packet queueing delay is dominated by the travel time of the MC, and at heavy loads (ρ approaching 1) it is dominated by the sojourn time of the MC at the sensors. With *exhaustive* and *gated* service, ρ should be less than 1 to ensure stability of the system. For stable symmetric systems with a single MC and same parameters, the mean waiting time of the packets is smallest with *exhaustive* service policy and largest with *1-Limited*. Also, ensuring stability with *1-Limited* service requires the mean travel time of the MC to be smaller than the service time, which is not practically feasible. *Exhaustive* service policy is the optimal one as far as the average packet delay is concerned.

However, the effectiveness of data collection can not be quantified using the mean waiting time of packets alone. Another important parameter that matters, especially in delay-sensitive environments, is the fairness of data collection. Under the assumption of symmetric queues with equal data generation rates, the mean waiting time is independent of sensor location and packets generated by all sensors receive same treatment. Now, let us consider a situation in which the packet generation rates differ considerably among sensors, resulting in unequal loads offered by them. Let $\rho_i = \lambda_i E[X]$ be the load at sensor i . By following the approach used in [36], we observe that the dependence of mean waiting time at sensor i on the load offered by node i under the *exhaustive*, *gated* and *1-limited* service policies, can be expressed, respectively as

$$E[WQ_i]_{exh} \propto (1 - \rho_i) \quad (12)$$

$$E[WQ_i]_{gated} \propto (1 + \rho_i) \quad (13)$$

$$E[WQ_i]_{lim} \propto (1 - \rho + \rho_i) \quad (14)$$

Equations (12), (13), and (14) reveal that the packets generated by different nodes are treated differently, based on the service policy. In *exhaustive* service, packets arriving to light-traffic sensors have longer average waiting time than those arriving to heavy traffic sensors. But in *gated* and *1-limited* service schemes, it is in the other way. The *1-limited* service policy is usually considered to be a fair policy since only one packet is collected from each sensor in a cycle of the MC. *Exhaustive* service is less fair since one heavily loaded sensor can dominate the system, and will occupy the MC for a long time. This means that, although the average waiting time may be smaller for *exhaustive* service compared to the other two service policies, the maximal waiting time at the lightly loaded sensors may be larger. Hence, though *exhaustive* service gives optimum performance for delay-tolerant applications, it is not the optimum one for delay-sensitive applications. In delay-sensitive applications, if the packets are not collected before their deadline or expiry time, they will have to be discarded, thus reducing the number of packets receiving on-time service.

Computation of the average queueing delay of packets using (9) requires the knowledge of the mean and variance of MC travel time. To evaluate these parameters, we follow an approach similar to that in [26]. In our system model, the BS maintains a queue to store the received requests and serve them according to its service discipline, the simplest one being first-come-first-served (FCFS). We assume a square sensing field with static sensor nodes uniformly distributed in the network. The arrival of data collection requests to the BS is assumed to be a Poisson process and the communication is assumed to be loss-less. Due to the assumption of uniform distribution of node deployment, the locations of data collection requests can be treated as random points in the square sensing field, based on which the travel time of the MC between two consecutive locations can be evaluated. The probability density function of the distance between two arbitrary points in a unit square is given by

$$f_D(d) = \begin{cases} 2d(\pi - 4d + d^2) & 0 \leq d \leq 1 \\ 2d[2\sin^{-1}(\frac{1}{d}) - 2\sin^{-1}\sqrt{1 - \frac{1}{d^2}} + 4\sqrt{d^2 - 1} - d^2 - 2] & 1 \leq d \leq \sqrt{2} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

Using this, if the MC moves at a constant velocity V , the mean and variance of the MC travel time between two arbitrary points in a unit square area can be obtained as $0.4555/V$ and $3.95/V^2$, respectively.

Once the mean queueing delay of the message has been determined, the expected response time of the message, being the sum of its queueing delay in the sensor buffer plus service time by the MC (packet transmission time) can be obtained as

$$E[T] = E[WQ] + E[X] \quad (16)$$

The sensor buffer occupancy of a tagged sensor will be maximum (equal to λ times the MC cycle time) when the MC approaches it for data collection and minimum (equal to zero for *exhaustive* service) when it leaves the sensor. The average sensor buffer occupancy

$$E[N_Q] = \lambda E[WQ] \quad (17)$$

and the number of messages in the system (in queue and in service)

$$E[N] = \lambda E[T] \quad (18)$$

The message latency and sensor buffer occupancy increase with the number of nodes N , packet arrival rate λ and the size of the deployment area, where as it decreases with the speed of the MC. However, the speed of the MC can not be increased beyond a limit (of the order of 20 m/s) due to practical limitations. Thus, the delay performance of the MC-based DTN scheme with a single mobile element is not at all comparable with that of ad hoc multihop network (of the order of several minutes for the former, while a few seconds for the latter). Correspondingly, the buffer requirement of static sensors is negligible in an ad hoc network, while it is considerably high in the MC-based scheme. Hence, the MC-based data collection approach, as such, is suitable only for delay-tolerant applications. Techniques to improve the delay performance so as to extend its suitability for delay-sensitive applications will be discussed in Section V.

C. Packet Delivery Ratio (PDR)

The Bulk Service Queueing model for mobility-assisted data collection as used by Jain et al. in [25] permits us to evaluate the success ratio of data collection. Here, the data generation and MC arrival processes are assumed to be renewal processes with average rates λ and μ respectively. It is also assumed that when an MC visits a sensor, no other sensor is near-by and contending for service. A maximum of K packets is transferred from the sensor to the MC in each visit. Data transmission does not incur any loss and the only loss (if any) is due to sensor buffer overflow.

Since a maximum of K packets are collected in one visit of the MC, the net service rate is $K\mu$. If the random variable Q represents the queue length at the MC arrival instant, the average of Q is used as a measure of the sensor buffer occupancy, which in turn, decides the Packet Delivery Ratio (PDR). As the service size is K packets, and if Q (queue length at MC arrival instant) is less than K then only Q packets are served, clearly

$$PDR_{MC} = \frac{\mu E[\min(K, Q)]}{\lambda} \quad (19)$$

Assuming *exhaustive* service policy, all the data generated and buffered so far is transferred when the MC visits the sensor. Hence the amount of data in the sensor buffer when the MC approaches it, will be the minimum of : (i) the amount of data generated in one cycle time of the MC, and (ii) the sensor buffer size. For Poisson data generation, the amount of data generated in an interval depends only on the length of the interval and hence the expected sensor buffer length becomes

$$E[Q] = \frac{\lambda}{\mu} \quad (20)$$

For a fixed service size K , $E[Q]$ increases with λ and decreases with μ .

In a stable system, with the assumption of large K and infinite buffer space, and on substituting the value obtained from Eqn. (20) into (19), we get the Packet Delivery Ratio to be 1 with this model. If the sensor buffer space or the service size K is not sufficiently large to accommodate the incoming traffic without buffer overflow, packets will be dropped and PDR is reduced. Hence the sensor buffer size SB and service size K should be designed such that no packet is lost due to buffer overflow, for a given data generation rate λ and the MC arrival rate μ . However, sensor buffer size SB is limited by the size and hardware cost of the sensor memory.

Assuming ideal channel, no MC failures, and sufficiently large buffer space to avoid buffer overflow, the PDR will be theoretically 1 for delay-tolerant applications. But practically, there exists a probability that a node is not visited by the MC within a specified time period or deadline. In such situations, the significance of the data may be lost if the application is delay-sensitive, or the data itself may be lost due to buffer overflow. Since these two factors limit the PDR in MC-based data collection, care is to be taken to ensure that the sensors are equipped with sufficient buffer space as dictated by the load conditions, and the buffered packets are collected before their significance is lost, in delay-sensitive applications.

In the case of ad hoc multi hop network, the PDR is dependent on the node density, since a contemporaneous source-to-sink path should exist for successful packet delivery. To investigate the impact of node density on PDR, we assume the use of Vector-based Forwarding (VBF) as the routing protocol and follow the approach similar to the one used in [13] with appropriate modifications for 2-dimensional deployment. Assuming N nodes each with transmission range r , uniformly deployed in a square area of side A , the density d of nodes in the network = $\frac{N}{A^2}$. Now, if B represents the radius of the routing pipe in VBF, and P_l represents the loss probability of packets, it can be shown that the probability of successful delivery of a packet over h hops

$$PDR_{ad hoc} = \left[1 - P_l^{\frac{1}{3}\pi B r^3 d^2}\right]^h \quad (21)$$

Equation (21) shows that, for a fixed packet loss probability P_l , probability of successful packet delivery increases with node density, width of routing pipe, and transmission range of sensor nodes in the ad hoc multi hop network that uses VBF. Increase in the width of routing pipe or the transmission range of sensor nodes will result in increased energy consumption, where as high node density is not feasible due to cost constraints and deployment restrictions. Thus, achieving a reasonably good delivery performance using ad hoc multi hop approach for event-driven data collection in sparse and energy-constrained environments is almost impossible. At the same time, probability of successful packet delivery is independent of node density in MC-based data collection, thus making it the better option for sparse and constrained networks, as far as successful data delivery is concerned.

V. PERFORMANCE ENHANCEMENT

In this section, we investigate techniques to improve the delay and delivery performance of the basic DTN

scheme with a single MC. Based on the study of latency performance in Section IV, we propose four techniques to reduce data collection latency and to enhance the support for delay-sensitive applications: (i) Use of multiple mobile collectors, (ii) Activity-based periodic polling, (iii) Hybrid architecture with both static and mobile sinks, and (iv) Hierarchical organization of mobile collectors. In the first technique, more than one mobile collectors are used, thus increasing the effective service rate, thereby reducing the message waiting time. In the second one, different priority is assigned to different nodes (based on data generation rate, traffic class, etc.), and the order and/or frequency of polling or visiting the static sensor nodes is modified to account for the differing activity conditions. The third technique uses a hybrid architecture of both ad-hoc multi-hop and mobility-assisted schemes, exploiting the advantages of both. In the fourth one, the MCs are organized in a hierarchical manner such that application-oriented differentiated packet delivery is made possible. We present these four techniques, as well as analytical expressions for the evaluation of their latency and delivery performance.

A. Multiple Mobile Collectors

In our basic polling model, there is only a single server, servicing a number of queues in a cyclic manner, which has been found to be unsuitable for delay-sensitive applications. When the input load is too high or the deadline requirements of the application are quite demanding, the BS may decide to schedule multiple MCs with different visit tables assigned to each. When the number of MCs is increased, the model is converted to a Multi Server Multi Queue (MSMQ) system or *multi server polling model*, the exact analysis of which is not available. Assuming independent MCs, symmetric Poisson-distributed data arrivals, independent and identically distributed *service times* and *walk times* and no server clustering, an approximate expression for the mean waiting time can be derived following the approach used in [37]. The total average amount of work arriving to the MSMQ per unit amount of time remains unchanged ($= N\lambda E[X]$) as in the single server system. At steady state, the MCs evenly share this load and if S is the number of MCs, the utilization factor of any one MC becomes

$$\rho_s = \frac{N\lambda E[X]}{S} \quad (22)$$

The time interval between two consecutive arrivals of any one MC at a tagged sensor buffer q can be evaluated as

$$E[C_q] = \frac{R}{S - N\lambda E[X]} \quad (23)$$

for $q = 1..N$.

Since stability is guaranteed by the finiteness of average cycle time, to ensure stability, the number of MCs

$$S > N\lambda E[X] \quad (24)$$

In other words, the packet arrival rate

$$\lambda < \frac{S}{NE[X]} \quad (25)$$

To get the mean message waiting time in the multiple MC case, the expression for mean waiting time in single MC

case as given by (9) can be modified by substituting $E[X]/S$, $E[X^2]/S^2$, $E[W]/[S-(S-1)\rho]$, and $E[W^2]/[S-(S-1)\rho]^2$ in place of, respectively, $E[X]$, $E[X^2]$, $E[W]$, and $E[W^2]$. Thus, the mean waiting time in the multiple MC situation becomes

$$E[W_q] = \frac{E[W^2]}{2E[W][S-(S-1)\rho]} + \frac{N \left[\frac{\lambda E[X^2]}{S} + \frac{E[W](S-\lambda E[X])}{S-(S-1)\rho} \right]}{2(S - N\lambda E[X])} \quad (26)$$

Similar to the basic single MC network, here also, the expected waiting time of the packet in the sensor buffer and the average sensor buffer occupancy increase with the packet arrival rate λ , number of sensors N and the size of the deployment area and reduces with the speed of MCs. However, both performance metrics decrease with the number of MCs S , thus making the model better suited for heavy input load conditions, memory-limited sensors, and delay-sensitive applications. While the delay and delivery performance are improved by the use of multiple data collectors, energy consumption and network lifetime are not affected, since the number of transmissions and the range of transmission are not changed by the use of more number of MCs. Taking into account the higher cost of MCs compared to ordinary sensor nodes, minimum number of MCs that satisfy the application-specific latency constraints may be used.

B. Activity-based Priority Polling

In practical situations, all the sensor nodes may not be generating data at the same rate and hence our earlier assumption of symmetric queues may not be valid always. More packets will be generated in some areas having high activity that require immediate attention, while some other areas may be generating very few packets only. In such situations, it will not be efficient and fair to visit all the sensors in a cyclic manner. When the data generation rates among the static sensor nodes vary considerably, it will be better to visit the nodes with higher arrival rates more frequently, rather than following the cyclic order. In cyclic polling, the server visits the queues in the order $Q_1, Q_2, \dots, Q_N, Q_1, Q_2, \dots, Q_N, \dots$. In *Periodic* polling, the server visits the queues in a fixed order specified by a *polling table* in which each queue occurs at least once [38].

Consider the single server polling model with the difference that the arrival rates at the queues are not equal, instead the packet arrival intensity at sensor i is λ_i , $i = 1, \dots, N$. The offered load at sensor i is $\rho_i = \lambda_i E[X_i]$, where $E[X_i]$ is the mean service time at sensor i . The total offered load in the network $\rho = \sum_{i=1}^N \rho_i$. The MC visits the sensors according to a periodic - not necessarily cyclic - polling scheme. The approach followed in [38] can be used to minimize the workload in the system and to ensure *fairness* among the sensors by using optimum visit frequencies. For *exhaustive* service, assuming W_i to be the switch-over time from queue $i-1$ to queue i , the visit frequency at node i becomes

$$f_i^{exh} = \frac{\sqrt{\rho_i(1-\rho_i)/W_i}}{\sum_{j=1}^N \sqrt{\rho_j(1-\rho_j)/W_j}} \quad (27)$$

Now, all the nodes are not visited equally in a cycle, instead the nodes having more buffered data waiting for transmission

(due to higher packet generation rate) will be visited more often than those with less buffered data. Assume that sensor i is visited n_i times in a cycle of the MC and these visits are spread as evenly as possible. Considering the interval between two successive MC visits to a node i as a sub cycle, the mean residual time of a sub cycle of i will be

$$ERSC_i \propto \frac{E[C]}{n_i} \quad (28)$$

where $E[C]$ is the mean time for one complete visit cycle of the MC according to the polling table. Now the mean waiting time at node i will be [38]:

$$(W_q)_i \propto (1 - \rho_i) \frac{E[C]}{n_i} \quad (29)$$

which shows that the sensor nodes with high data generation rates (having high values of ρ_i and n_i) get better treatment and majority of the generated packets get good treatment, in terms of waiting time and buffer requirement.

C. Hybrid Architecture

The scheme of employing a hybrid architecture of both ad hoc multi-hop and mobility-assisted data collection approaches is proposed here to provide application-oriented packet delivery. As illustrated in Fig. 3, there exist both static and mobile sinks in the network, the former one for collecting delay-sensitive critical data and the latter one for delay-tolerant bulk data. While setting up the network, the static sensors are organized in to a number of routing trees rooted at the static sink located at the centre of the deployment area. Mobile sink (MS) covers the entire network by following a trajectory suitable for periodic data collection, as shown in the diagram. It sojourns at predefined locations so as to collect buffered data from the near by sensors. The number of sojourn points and the transmission range of sensor nodes can be adjusted according to the node energy constraints and the message deadline requirements.

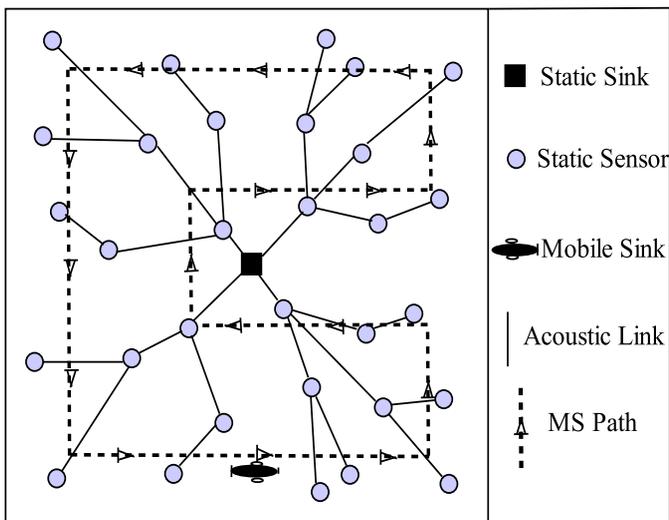


Figure 3. Hybrid architecture with static and mobile sinks

Each sensor node always keeps up to date information about its static path to the static sink. Any emergency

information that requires immediate attention like disaster warning, hazard detection, etc., is transferred to the static sink using the ad hoc multi hop path, provided such an end-to-end contemporaneous path exists. If not, the sensor has to either transmit at a higher power level to improve connectivity at the expense of increased energy consumption. If that too fails, the node has to wait for the arrival of the MC. Because ad hoc multi-hop communication is used only for applications with tight deadline requirements, the sink neighbourhood or hot-spot problem is not expected to be as severe as in the pure ad hoc multi-hop approach. Hybrid architecture permits us to achieve trade-off between network lifetime and timeliness of data collection, though with increased complexity of keeping two data collection approaches in the same network.

The performance metrics like energy consumption, energy balancing, packet latency, and sensor buffer occupancy depend on the type of communication (ad hoc multi-hop or mobility-assisted), which in turn, depends on the nature of application: delay-sensitive or delay-tolerant. For delay-tolerant applications, mobility-assisted data collection with a single MC is made use of. The MC visits the sensors as discussed in the basic model in Section III for event-driven data collection or by following a trajectory as illustrated in Fig. 3, covering the entire deployment area in a cyclic fashion for periodic data collection. For symmetric queues, the delay performance of data collection can be evaluated using (9), (10), (16), and (17). For delay-sensitive applications, ad hoc multi-hop connection to the static sink is used, whose delay is negligible compared to MC-based approach, thus ensuring timely delivery of emergency data. However, for successful packet delivery, end-to-end connectivity is to be ensured, failing which packets will be dropped. We use VBF protocol [13] for multi hop routing. In VBF, the PDR is dependent on the density of nodes, width of routing pipe and transmission range of sensor nodes.

D. Hierarchical Architecture

This architecture is suitable for large networks with periodic and event-driven data collection, supporting both delay-tolerant and delay-sensitive applications. Unlike the hybrid architecture, there is no static sink here. The static nodes as well as the mobile collectors are organized into a number of tiers forming a hierarchical architecture. We have considered three tiers, which can be extended further according to the size of the network and the requirements of the application. As illustrated in Fig. 4, the entire network is organized into four non-overlapping clusters and three hierarchical tiers. All static sensor nodes have the basic responsibility of sensing the environment and buffering the sensed data. Additionally, they selectively forward delay-sensitive traffic to nodes that are more frequently visited by the MC.

The entire network area is divided into 16 equal square partitions (not shown in the diagram). During the set up phase of each round of data collection, Tier-3 nodes in each partition select one of them as a Tier-2 node based on residual energy; and Tier-2 nodes belonging to each cluster select one of them as a Tier-1 node based on proximity to the centre. Hence, there will be a maximum of 16 Tier-2 nodes and four Tier-1 nodes in the network under consideration. Tier-1 MC (MC1) cycles among the Tier-1 nodes alone, while the Tier-2 MC (MC2) cycles among the Tier-2 nodes. Tier-3 MC (MC3) follows a

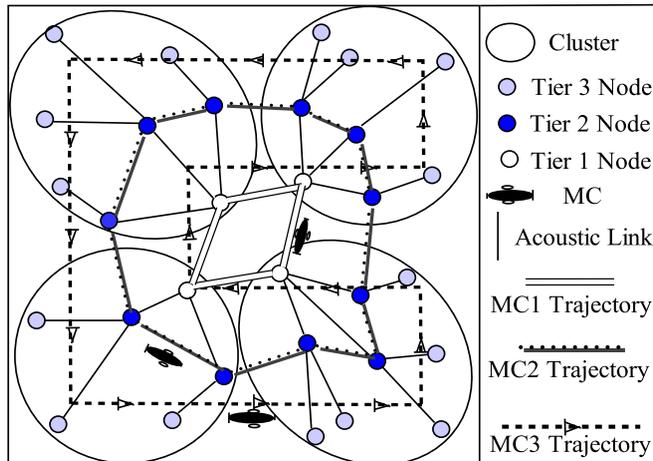


Figure 4. Hierarchical architecture

trajectory passing through the centres of all the 16 partitions, sojourning at the 16 locations for collecting delay-tolerant data (if any) from the Tier-3 nodes. Cycle time of the MC1 will be very small compared to its counterparts in other tiers. In other words, Tier-1 nodes are visited more often than Tier-2 nodes, while the latter is visited more often than Tier-3 nodes.

Based on the urgency of the sensed data, packets may be buffered at the originating node or forwarded to a node in the next tier. Upon receiving a packet, a Tier-3 node will check its delay-sensitivity. If it is delay-tolerant, it will be stored in the sensor buffer, to be collected by MC3. Otherwise, the packet will be forwarded to the respective Tier-2 node, where it will be buffered, to be collected by MC2. Tier-2 node will check whether the packet is time-critical and if so, it will be immediately forwarded to the Tier-1 node. Since the Tier-1 nodes are visited quite frequently, the latency performance will be good. Data is assumed to have been successfully delivered once it has been collected by any one MC.

For performance evaluation, we consider a square deployment area of size $2000\text{m} \times 2000\text{m}$ and velocity of all the MCs to be 15 m/s . MC3 follows a trajectory as shown in Fig. 4 and for analytical tractability, trajectories of MC2 and MC1 are approximated by square paths of side 1000m and 500m , respectively, around the centre of the deployment area. Packet generation is assumed to be Poisson and symmetrical, with 10% of the generated packets being time-critical and 30% of the generated packets being delay-sensitive, but not time-critical. The maximum and average waiting times of each category of packets can be evaluated using (9) and (10). Unlike the hybrid architecture, here no multi-hop communication is used and hence PDR is independent of node density.

VI. ANALYTICAL AND SIMULATION RESULTS

Extensive simulations were done to validate our analytical results using the NS-2 based network simulator for underwater applications, Aqua-Sim. The unique characteristics of UWSNs like acoustic attenuation model, acoustic channel model, 3-dimensional deployment and very slow propagation make the existing terrestrial network simulators unsuitable for UWSN simulation study and resulted in the development

of Aqua-Sim. Aqua-Sim is an event-driven, object-oriented simulator written in C++ with an OTCL (Object-oriented Tool Command Language) interpreter as the front-end. Following the object-oriented design style of NS-2, all UWSN entities are implemented as classes in C++. Several interesting works like [39] and [40] have already been implemented in this simulation system.

The codes simulating underwater sensor nodes, traffic, acoustic channels, MAC protocols, and a few routing protocols are already available in Aqua-Sim. We have incorporated in it, the DTN concepts of beaconing, *contact* discovery and *store-carry-and-forward* and the polling based (*exhaustive* service) data collection. Energy model with tunable transmit power and latency minimization techniques like the use of multiple mobile collectors, visit-frequency based priority polling, hybrid architecture with static and mobile sinks, and the hierarchical organization of sensors and mobile collectors were also implemented.

We have used the VBF routing protocol for implementing the multi hop network for comparison purpose and the MC-based DTN protocol developed by us in Aqua-Sim for the mobility-assisted short-range data collection purpose. We have employed the *Broadcast MAC* protocol with carrier sensing and collision avoidance, in which the MAC first senses the channel when a node has packet to send. If the channel is found to be free, the node broadcasts the packet, otherwise it backs off. If the number of back off exceeds a limit, the packet is discarded.

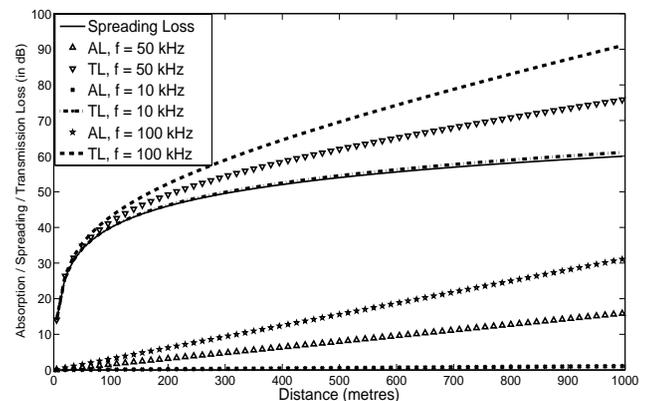


Figure 5. Transmission Losses in the deep water scenario (Analytical): AL - Absorption Loss, TL - Transmission Loss

Fig. 5 illustrates the impact of frequency of operation and distance between the sensor nodes on the total transmission loss in deep water, as expressed by (2). We have assumed a target SNR of 20 dB and noise level of 70 dB for this result. Transmission loss is the sum of spreading loss and absorption loss. Spreading loss is independent of frequency and its variation with distance is quadratic in deep water. Absorption loss increases with frequency and distance between nodes.

Higher transmission loss at larger source-to-sink distance leads to increased energy consumption as illustrated in Fig. 6 for the deep water scenario. Assuming tunable transmit power P_t , receive power P_r fixed at 0.075 W , and packet length L

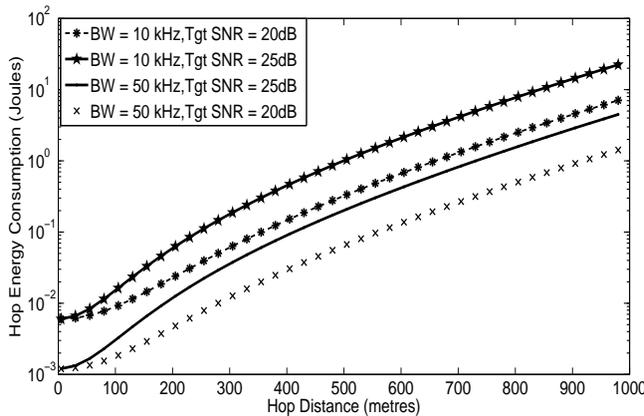


Figure 6. Hop Energy Consumption for varying hop length and bandwidth

fixed to 400 bits [9], the effect of hop length, target SNR, and channel bandwidth on per-hop energy consumption as expressed by (4) is plotted here. Decreasing the source to sink distance reduces the transmission loss and increasing the bandwidth reduces the time required for transmission. Both situations lead to reduced transmit energy consumption, thus validating the suitability of short range communication in energy-constrained environments.

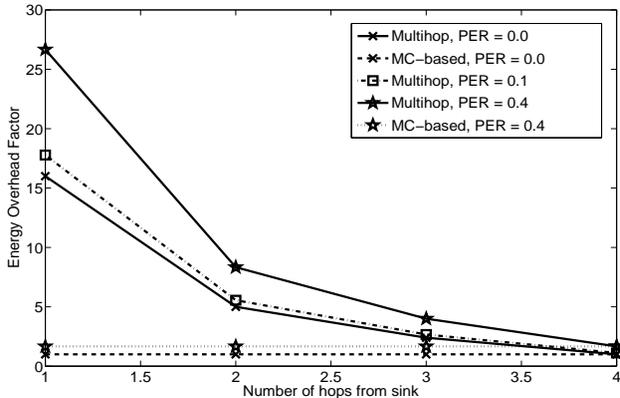


Figure 7. Transmit Energy Overhead of static sensor nodes with multi-hop and MC-based schemes for different PERs

Assuming static sensor nodes having transmission range 250m uniformly distributed in a circular area of radius 1000m, a comparison of the *Energy Overhead Factor* (defined in Section IV.A) in mobility-assisted and ad hoc multi-hop approaches is illustrated in Fig. 7. The variation of *EOF* of a node with its proximity to sink is also shown. As expected, nodes in the mobility-assisted approach have reduced and balanced overhead, irrespective of their location relative to the sink. At the same time, the relaying overhead of a sensor increases with its proximity to sink in the ad hoc multi hop network. The impact of packet error rate (PER) on the energy overhead due to non-ideal channel is also shown in this figure. Due to increased relaying overhead, the nodes nearer to the sink will deplete their battery power soon. If we define the lifetime of a network as the timespan till the first node dies

due to energy depletion, it is evident that the use of mobile elements for data collection leads to enhanced lifetime of the network due to reduced and balanced energy consumption among the sensor nodes.

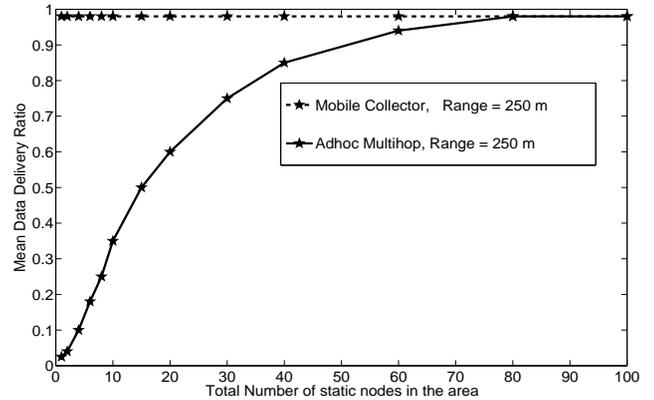


Figure 8. PDR with multi-hop and MC-based data collection (Delay-tolerant application)

The variation of PDR with node density is shown in Fig. 8. Assuming infinite buffer size and no communication errors, ideally the PDR should be 1 for the MC-based data collection scheme, irrespective of the number of nodes in the network. For ad hoc multi-hop network, as indicated by (21), delivery ratio is very small for low node density due to end-to-end connectivity issues. As the node density is increased, PDR increases initially and finally reaches a maximum value and then remains almost constant. For the MC-based scheme, delivery ratio is independent of node density. Hence, it is the ideal one for sparse and disconnected networks, provided the network lifetime and successful data delivery are of prime concern and the application is not time-critical. If the sensors are not equipped with sufficient buffer space to avoid buffer overflow at high loads, packets will be dropped and PDR reduced. Also, in delay-sensitive applications, if the packets are not received before the application-specified deadline, significance of the data will be lost, which is equivalent to loss of packets that leads to reduced PDR.

Assuming controlled motion of a single MC with speed 15 m/s for on-demand data collection in a square area of size 2000m × 2000m with 10 nodes randomly and uniformly distributed in this area, analytical results illustrating the variation of mean waiting time of a packet, mean cycle time of MC, mean travel time of the MC in a cycle, mean sojourn time of the MC in a cycle and the inter-visit time of the MC at a tagged sensor node were obtained for varying load conditions using our basic model and plotted in Fig. 9. The sensors are visited by the MC based on FCFS policy as indicated by the visit table assigned to it and the sensor buffers are serviced according to the *Exhaustive* service policy. As expected, the waiting time of packets, cycle time of the MC, sojourn time of the MC in a cycle, and the inter-visit time at a tagged node increase with the system load. However, the walk time of the MC is independent of the load. At light loads, the cycle time of the MC and the waiting time of the packets are dominated by the travel time of the MC, while at heavy loads, they are dominated by the sojourn time (pause time of the MC near

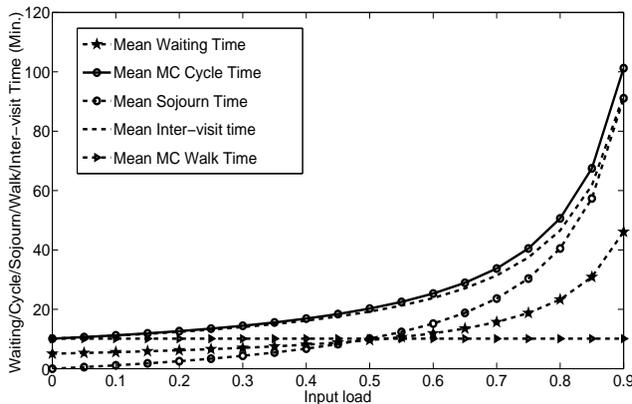


Figure 9. Mean packet waiting time / Cycle time /Data Transfer Time / Inter-visit Time with polling model (Analytical)

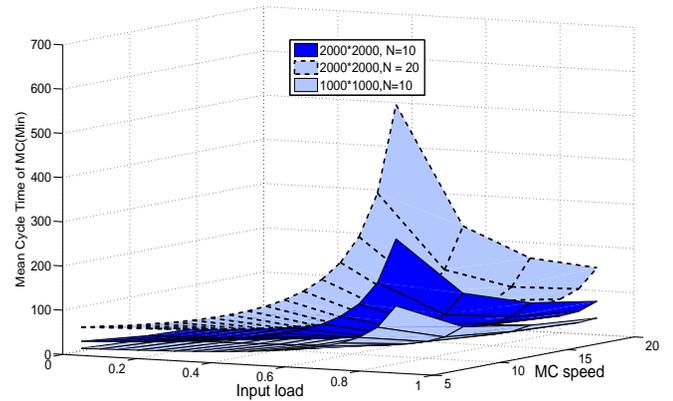


Figure 11. Mean Cycle time variation with load, speed and deployment area (Analytical)

the sensor for data transfer). When the system load approaches unity, stability is lost and the delay values grow exponentially. This situation should be avoided, otherwise delay will not be bounded and sensor buffers will overflow. The results also act as a guide to decide when to go for multiple mobile collectors for meeting the delay constraints specified by the application.

behaviour. The buffer occupancy is zero when the MC leaves a sensor and maximum when the MC approaches it. If the sensor buffer space is not sufficiently high, packets will be lost due to buffer overflow, resulting in reduced PDR.

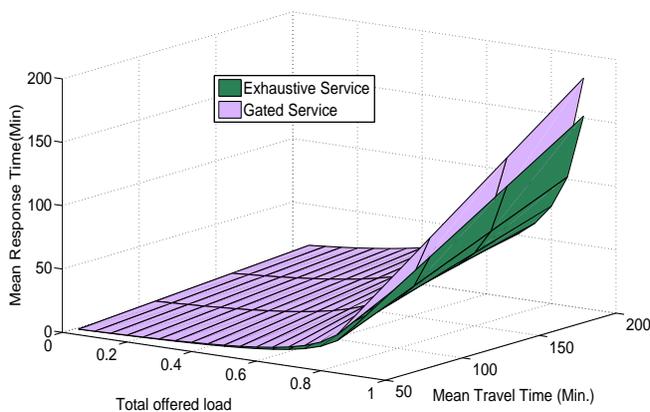


Figure 10. Mean Response Time with different service policies (Analytical)

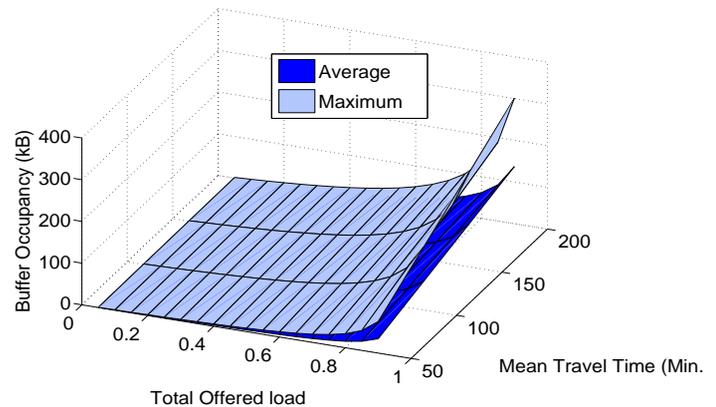


Figure 12. Variation of sensor buffer occupancy (Analytical)

The variation of mean response time of a packet, as evaluated using (16) for varying input load under the *exhaustive* and *gated* service policies, is illustrated in Fig. 10. Compared to *gated* service policy, *exhaustive* service policy results in smaller mean waiting time and response time, and hence more optimal. For a fixed input load, response time increases with the mean travel time of the MC. Hence, efficient MC scheduling policies can be employed to reduce the travel time and to improve the latency performance of data collection.

Assuming *exhaustive* service policy of the MC, the impact of factors like input load, MC speed, number of sensors and dimensions of the sensor deployment area on the average cycle time of the MC is illustrated in Fig. 11. As expected, the MC cycle time increases with the number of nodes and area of deployment, whereas it decreases with MC speed. The sensor buffer occupancy, shown in Fig. 12 also exhibits a similar

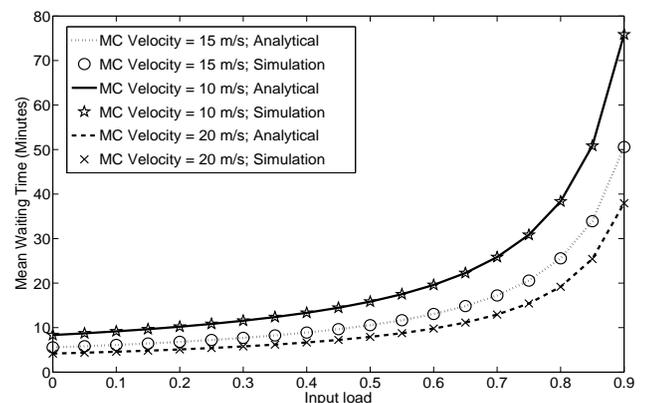


Figure 13. Waiting Time variation with load and MC speed

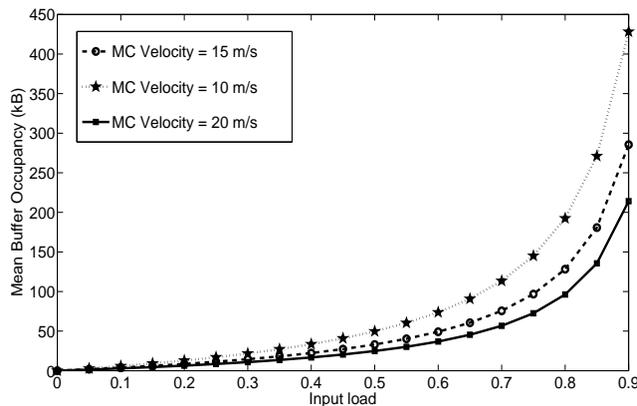


Figure 14. Variation of Mean Buffer Occupancy with speed of MC

Simulations were done with the same network conditions and fixing the packet size at 50 Bytes and data rate at 10 Kbps. The mean waiting time obtained for different values of data generation rate and different speeds of the single MC are plotted in Fig. 13. The sensors are equipped with sufficient buffer space so that packets are not lost due to buffer overflow. For a fixed number of nodes, deployment area, packet size, and MC speed, the variation in input load is effected by varying the packet generation rate. The mean waiting time increases with the input load and decreases with the speed of the MC. Analytical and simulation results show close agreement, validating the suitability of our model.

The analytical results showing the impact of input load and MC speed on the mean sensor buffer occupancy in the basic framework has been illustrated in Fig. 14. Similar to the waiting time of packets, the average number of packets in the sensor buffer awaiting their turn for transmission also increases with input load and decreases with MC speed. This result gives us an idea about how to decide the buffer size of the sensors, considering the load conditions and MC speed.

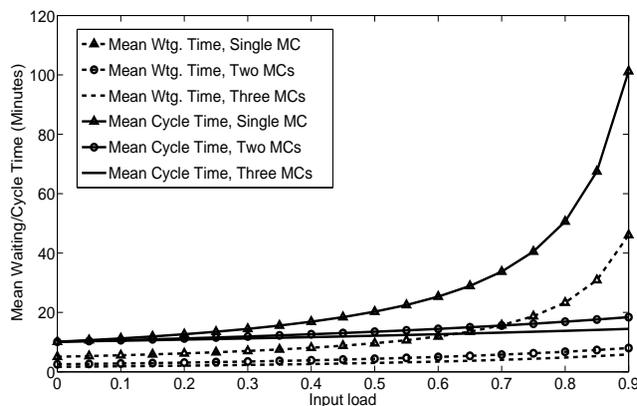


Figure 15. Delay Performance with Multiple MCs

Since it is not practical to have MC speeds above 20 m/s, use of multiple MCs is to be adopted for heavy traffic environments, delay-sensitive applications, and very limited sensor buffer situations. Keeping the same network conditions

as used in our basic model, the results showing the impact of number of MCs on packet delay performance is plotted in Fig. 15. As expected, for a fixed number of MCs, the mean waiting time increases with input load. Also, for a fixed load and MC speed, as the number of MCs is increased, the queuing delay is decreased. The sensor buffer occupancy also shows the same behaviour, as illustrated in Fig. 16. Simulation results related to mean waiting time and sensor buffer occupancy have shown close agreement with the analytical ones.

However, the cost of MCs is much larger compared to that of ordinary sensor nodes and a large number of MCs will lead to interference problems and increased complexity of implementation. Hence, the optimum number of MCs may be selected based on the delay constraints of the application and the cost considerations of MC deployment. In the scenario we have considered, it is observed that the performance gain obtained by using 3 MCs over 2 MCs is much less compared to that obtained by using 2 MCs over a single one. The results act as a guide to decide the number of MCs, based on the application-specified delay constraints and sensor-specific buffer space constraints.

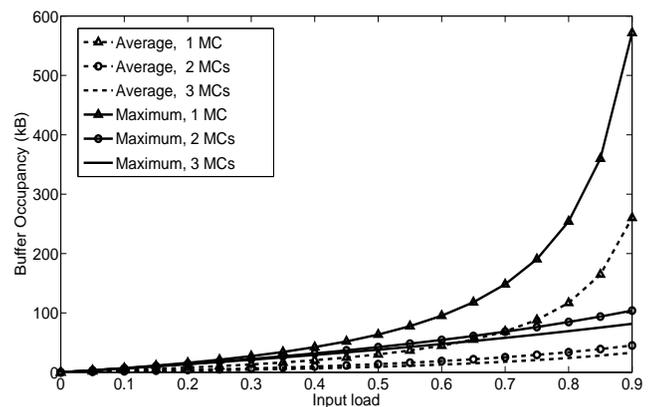


Figure 16. Sensor Buffer Occupancy with Multiple MCs

To demonstrate the improvement in delay performance due to activity-based priority polling, we have considered the same network conditions with 10 sensor nodes randomly and uniformly distributed in a square area of size $2000m \times 2000m$, generating packets (of size 50 bytes) at different rates, a single MC moving at 15 m/s, and the channel having a data rate 10 kbps. Table I gives the visit frequency and the mean waiting time for different packet arrival rates. Based on simulation for a fixed finite amount of time, the packets failing to get service due to the single MC not arriving within a fixed time interval is also noted.

With same packet size and data transmission rates, the nodes with high packet generation rate contribute more to the system load. In our activity-based priority polling scheme, nodes with higher load receive preferential treatment in terms of number of visits in a single cycle of the MC. The more the frequency of MC visits at a sensor node, the less the queuing delay of packets in the buffer and higher the chance of being collected before their application-specified deadline. In the scenario we have considered, node G generates 8 packets per minute, and these packets experience an average queuing

TABLE I. MEAN MESSAGE WAITING TIME AND PDR AT DIFFERENT NODES

Node id	Arrival Rate (Pkts/min)	MC Visits	PDR (%)	Waiting Time (Minutes)
A	0.01	3	16.7	23
B	0.1	10	33.3	18.7
C	0.25	17	38.2	11.5
D	0.5	24	46.3	10.7
E	1.0	34	54.5	8.78
F	4.0	69	91.1	8.1
G	8.0	96	99.6	7.0
H	0.2	15	35.7	13.9
I	5.0	77	92.6	7.5
J	2.0	49	74.4	8.3

delay of 7 minutes. At the same time, node *A* generates only 0.01 packets per minute and they experience an average queueing delay of 23 minutes. Similarly, fixing the packet collection deadline at 7 minutes, very few (only 0.4 %) packets generated by the highly active node *G* miss the deadline. Though the deadline miss ratio is high (83.3 %) in the case of node *A*, *A* can naturally be assumed to be placed in a relatively inactive region of the network generating very few packets, the successful collection of which does not contribute considerably to the overall functioning of the network. Thus, the scheme provides support for delay-sensitive applications and differentiated packet delivery, by reducing the packet waiting time and deadline miss ratio in the areas of high event activity.

Due to unequal visit frequency at different nodes, each node receives its share of service from the MC proportional to its sensing activity or the load offered by it. In addition, by reducing the unnecessary travels of MC to the low data rate regions in the network, the overall system utilization is improved and the fraction of packets getting collected within the specified deadline is increased. However, this performance enhancement is at the cost of increased waiting time and deadline misses at the low load nodes. Thus, though the scheme ensures fairness by means of allocating service of MC proportional to the offered load, the scheme appears to be not fair in terms of the mean waiting time and deadline miss ratio at all nodes in the network.

The results demonstrating the impact of using a hybrid architecture for data collection in delay-sensitive applications are illustrated in Fig. 17. As expected, the PDR for delay-sensitive applications depends heavily on the node's end-to-end connectivity with the static sink, which in turn depends on the density of nodes in the area and the transmission range of nodes. As the network becomes sparse, connectivity gaps occur and sensors get isolated from sink, resulting in increased deadline miss ratio. The option available is to communicate at a higher transmit power for critical situations, of course at the expense of increased energy consumption. Thus, the results exhibit two trade offs: (i) between the probability of on-time service completion and cost of deploying large number of sensor nodes; and (ii) between the probability of on-time service completion and energy consumption due to higher transmission range. Delivery

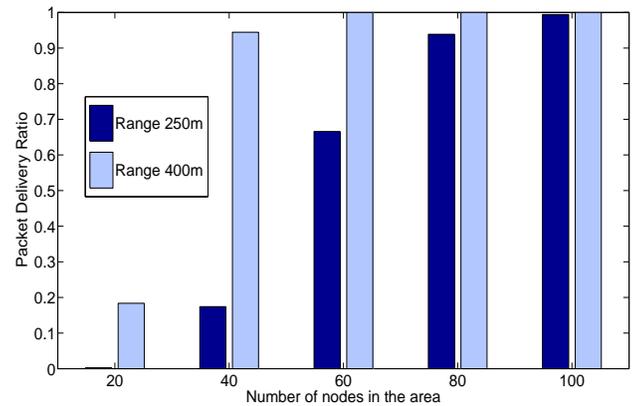


Figure 17. PDR for Delay-sensitive Applications using Hybrid Architecture

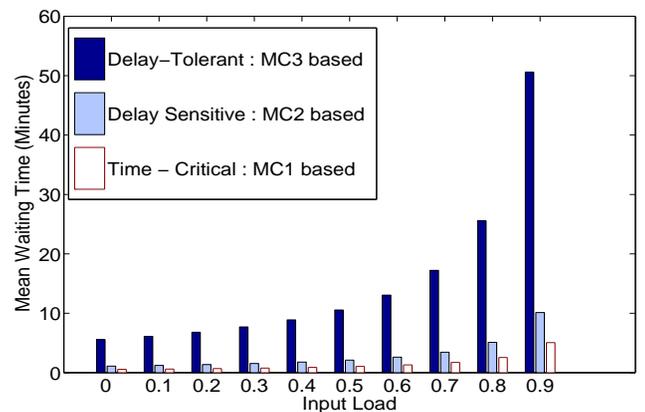


Figure 18. Delay performance using Hierarchical Architecture

performance for delay-tolerant applications are similar to that of the basic MC-based model. Latency performance is not shown, since the latency of delay-sensitive packets is negligible due to the use of ad hoc multi hop approach and that of delay-tolerant packets has already been discussed.

The latency performance of the architecture with hierarchical organization of sensor nodes and MCs is illustrated in Fig. 18. Packets belonging to different applications receive differential treatment in terms of average waiting time. Since good latency performance implies good delivery performance in MC-based delay-sensitive data collection, packet miss ratio will be minimum for all the three types of applications supported: Delay-tolerant, Delay-sensitive and Time-critical. Compared to the hybrid architecture, the advantage is the independence of delivery performance on node density. However, the scheme suffers from increased complexity of maintaining 3 MCs with hierarchical organization.

VII. CONCLUSION

Application-oriented event-driven data collection in sparse underwater acoustic sensor networks has been investigated in this paper. First, a mobility-assisted framework for energy-efficient on-demand data collection for delay-tolerant

application has been proposed. Analytical models for performance metrics like energy efficiency, message latency, packet delivery ratio, and sensor buffer requirement have been evaluated. The basic mobility-assisted data collection framework for event-driven data collection has been found to exhibit superior performance over ad-hoc multi-hop network in terms of energy efficiency and packet delivery ratio at the cost of increased latency. Thus, it is more suited for sparse or disconnected networks and in situations where network lifetime is more important than message delay.

We have augmented the basic model with techniques for improving the latency performance so as to support delay-sensitive applications also. Techniques like multiple mobile collectors and activity-based priority polling have been found to improve the delay and delivery performance. Hybrid architecture with static and mobile sinks as well as the hierarchical architecture of mobile collectors have been proposed to support application-oriented differentiated packet delivery. The basic DTN framework for delay-tolerant applications and the enhanced models for delay-sensitive applications have been implemented in the NS-2 based network simulator, thus enhancing the scope for further research in this area. As a future work, we plan to extend this study to 3-dimensional networks and to investigate techniques for optimizing the network performance adaptively based on application requirements and network constraints.

REFERENCES

- [1] J. M. Janardanan Kartha and L. Jacob, "On-demand data collection in sparse underwater acoustic sensor networks using mobile elements," in Proceedings of The Tenth International Conference on Wireless and Mobile Communications, ICWMC 2014, Seville, Spain, pp. 105–111.
- [2] M. Ardid, J. A. Martínez-Mora, M. Bou-Cabo, G. Larosa, S. Adrián-Martínez, and C. D. Llorens, "Acoustic transmitters for underwater neutrino telescopes," *Sensors*, vol. 12, no. 4, 2012, pp. 4113–4132.
- [3] S. Adrián-Martínez, M. Bou-Cabo, I. Felis, C. Llorens, J. Martínez-Mora, M. Saldaña, and M. Ardid, "Acoustic signal detection through the cross-correlation method in experiments with different signal to noise ratio and reverberation conditions," in *Ad-hoc Networks and Wireless*. Springer, 2014, pp. 66–79.
- [4] Uwsn research lab, university of connecticut. [Online]. Available: <http://uwsn.engr.uconn.edu/>
- [5] Woods hole oceanographic institution. [Online]. Available: <http://www.whoi.edu/>
- [6] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad hoc networks*, vol. 3, no. 3, 2005, pp. 257–279.
- [7] K. Fall, "A delay-tolerant network architecture for challenged internets," in Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003, pp. 27–34.
- [8] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, no. 2, 2003, pp. 215–233.
- [9] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. Cui, Z. Shi, Y. Fei, and S. Zhou, "Aqua-sim: an ns-2 based simulator for underwater sensor networks," in OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges. IEEE, 2009, pp. 1–7.
- [10] ns-2 network simulator. [Online]. Available: <http://www.isi.edu/nsnam/ns>, [last accessed : May 2015]
- [11] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, 2011, pp. 1908–1927.
- [12] S. Ahmed, I. Khan, M. Rasheed, M. Ilahi, R. Khan, S. H. Bouk, and N. Javaid, "Comparative analysis of routing protocols for under water wireless sensor networks," arXiv preprint arXiv:1306.1148, 2013.
- [13] P. Xie, J.-H. Cui, and L. Lao, "Vbf: vector-based forwarding protocol for underwater sensor networks," in NETWORKING 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. Springer, 2006, pp. 1216–1221.
- [14] X. Peng, Z. Zhong, N. Nicolas, C. Jun-Hong, and S. Zhijie, "Efficient vector-based forwarding for underwater sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010, pp. 1–13.
- [15] M. C. Domingo and R. Prior, "Energy analysis of routing protocols for underwater wireless sensor networks," *Computer Communications*, vol. 31, no. 6, 2008, pp. 1227–1238.
- [16] M. Zorzi, P. Casari, N. Baldo, and A. F. Harris, "Energy-efficient routing schemes for underwater acoustic networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 9, 2008, pp. 1754–1766.
- [17] K. S. Geethu and A. V. Babu, "Minimizing the total energy consumption in multi-hop uwasns," *Wireless Personal Communications*, 2015, pp. 1–17.
- [18] N. Javaid, M. Jafri, S. Ahmed, M. Jamil, Z. Khan, U. Qasim, and S. Al-Saleh, "Delay-sensitive routing schemes for underwater acoustic sensor networks," *International Journal of Distributed Sensor Networks*, 2014.
- [19] N. Javaid, M. R. Jafri, Z. A. Khan, N. Alrajeh, M. Imran, and A. Vasilakos, "Chain-based communication in cylindrical underwater wireless sensor networks," *Sensors*, vol. 15, no. 2, 2015, pp. 3625–3649.
- [20] S. Jain, K. Fall, and R. Patra, Routing in a delay tolerant network. ACM, 2004, vol. 34, no. 4.
- [21] S. Tolba, M. Hakami, A. Mithdir, Y. Zhu, S. Le, and J. H. Cui, "Underwater delay tolerant routing in action," in *Oceans*, 2012. IEEE, 2012, pp. 1–6.
- [22] M. J. Jalaja and L. Jacob, "Adaptive data collection in sparse underwater sensor networks using mobile elements," in *Ad-hoc Networks and Wireless*. Springer, 2014, pp. 57–65.
- [23] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2004, pp. 187–198.
- [24] Z. Guo, G. Colombo, B. Wang, J.-H. Cui, D. Maggiorini, and G. P. Rossi, "Adaptive routing in underwater delay/disruption tolerant sensor networks," in *Wireless on Demand Network Systems and Services*, 2008. WONS 2008. Fifth Annual Conference on. IEEE, 2008, pp. 31–39.
- [25] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting mobility for energy efficient data collection in wireless sensor networks," *Mobile Networks and Applications*, vol. 11, no. 3, 2006, pp. 327–339.
- [26] L. He, Y. Zhuang, J. Pan, and J. Xu, "Evaluating on-demand data collection with mobile elements in wireless sensor networks," in *Vehicular technology conference Fall (VTC 2010-Fall)*, 2010 IEEE 72nd. IEEE, 2010, pp. 1–5.
- [27] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 8, 2006, pp. 958–973.
- [28] S. Yoon, A. K. Azad, H. Oh, and S. Kim, "Aurp: An auv-aided underwater routing protocol for underwater acoustic sensor networks," *Sensors*, vol. 12, no. 2, 2012, pp. 1827–1845.
- [29] G. A. Hollinger et al., "Underwater data collection using robotic sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, 2012, pp. 899–911.
- [30] Y.-S. Chen and Y.-W. Lin, "Mobicast routing protocol for underwater sensor networks," *Sensors Journal, IEEE*, vol. 13, no. 2, 2013, pp. 737–749.
- [31] S. Motoyama, "Flexible polling-based scheduling with qos capability for wireless body sensor network," in *Local Computer Networks Workshops (LCN Workshops)*, 2012 IEEE 37th Conference on. IEEE, 2012, pp. 745–752.

- [32] V. Kavitha and E. Altman, "Queuing in space: Design of message ferry routes in static ad hoc networks," in *Teletraffic Congress, 2009. ITC 21 2009. 21st International*. IEEE, 2009, pp. 1–8.
- [33] J. Janardanan Kartha and L. Jacob, "Delay and lifetime performance of underwater wireless sensor networks with mobile element based data collection," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.
- [34] R. J. Urick, "Principles of underwater sound," McGraw-Hill, New York, London, 1983.
- [35] H. Takagi, *Stochastic analysis of computer and communication systems*. Elsevier Science Inc., 1990.
- [36] Y. Gong and R. De Koster, "A polling-based dynamic order picking system for online retailers," *IIE Transactions*, vol. 40, no. 11, 2008, pp. 1070–1082.
- [37] M. Ajmone Marsan, L. De Moraes, S. Donatelli, and F. Neri, "Cycles and waiting times in symmetric exhaustive and gated multiserver multiqueue systems," in *INFOCOM'92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies*. IEEE, 1992, pp. 2315–2324.
- [38] O. J. Boxma, *Analysis and optimization of polling systems*. Centre for Mathematics and Computer Science, 1991.
- [39] T. Hu and Y. Fei, "Qelar: a machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 6, 2010, pp. 796–809.
- [40] M. Xu, G. Liu, and J. Guan, "Towards a secure medium access control protocol for cluster-based underwater wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

The Challenge of On-demand Routing Protocols Improvement in Mobility Context

Tiguiane Yélérou*, Philippe Meseure[§], Anne-Marie Poussard[§], and Toundé Mesmin Dandjinou*

*Polytechnic University of Bobo-Dioulasso, Burkina Faso

[§]University of Poitiers, XLIM-SIC CNRS Lab, France

E-mail : tyelemou@gmail.com, (Philippe.Meseure, anne.marie.poussard)@univ-poitiers.fr, dandjimes@yahoo.fr

Abstract—Mobile Ad hoc NETWORKS are characterized by rapid change in their topology and the lossy nature of wireless links. In this context, achieving respect the QoS constraints of multimedia communications is a real challenge. Routing protocols play an important role in achieving the required performance. They should be smart enough to select better paths for data transmissions. Several routing approaches are used. Most are derived from adapting those used in wired networks. They are not suited to the context of wireless networks. We focus on the well-known Ad-hoc On-demand Distance Vector (AODV) protocol, commonly used protocol in ad hoc network. Standard route discovery process used in AODV is expected to obtain the best path in term of delay. However, in lossy-links context, multimedia data packet transmission success, on path established thanks to control packets, may require several attempts. These retransmissions increase delay and overhead. Many QoS-based methods failed to make a meaningful improvement due to added complexity and additional delay and overhead. In this paper, we use a convenient and practical way to evaluate quality of links in mobile context. With these measures relating to the number of retransmissions, we produce a new metric. Thereafter, we use this metric to highlight the limitations of QoS approaches used to improve the performance of AODV. We show that improving performance of on-demand routing protocols, in the mobility context, lies on effective control of node neighborhood.

Keywords—mobility; reliability; wireless networks; quality of service; on-demand routing.

I. INTRODUCTION

In the socio-economic context marked by the need to communicate any time and any where, use of Mobile Ad hoc NETWORKS (MANET) in the communication chain is essential. However, to communicate in these networks with a satisfactory level of Quality of Service (QoS) stands as a challenge to network services and infrastructures developers. Due to the unstable nature of radio links, routing protocols that establish communication path, are struggling to find and to maintain the best paths between a source and a destination. Indeed, due to the mobility of nodes, obstacles in the medium of the radio wave propagation and interference, the radio links are broken quickly.

In order to guarantee QoS, routing protocols should be smart enough to choose a reliable route in order to avoid packet loss. To deal with the problem, QoS-based routing protocols are proposed. Route selection process should take into account link quality. However, most methods proposed for link quality estimation and best path selection are not appropriate for

this rapid topology change. Most QoS protocols have many problems, among which, we can mention the additional costs induced by the determination of the value of the metric used, the accuracy of the value of the metric used, the additional complexity made to the route selection process. The obtained path is, very often, longer than the shortest path (in terms of number of hops). In mobility context, long paths are more vulnerable to breakage than shortest paths. In [1], a primary study was conducted on these issues.

In this paper, we use a realistic simulation environment to explain the problem. We use of a convenient and practical way to evaluate quality of links in mobile context. Then, we design different QoS-based Ad-hoc On-demand Distance Vector (AODV) protocols. The QoS metric used (called PR-metric) is based on the number of retransmissions. It takes into account accurately the proportion of retransmission time with respect to time of first issue. We use this metric to compare effectiveness of different QoS-based methods used to improve on-demand routing protocols performance. Finally, we conduct a detailed analysis of differend QoS-based AODV protocol performance.

The remainder of the paper is organized as follows: In Section II, we present and analyze related work. In Section III, we present our QoS-based routing protocols. Performance evaluation and discussions are made in Section IV. We conclude in Section V.

II. RELATED WORK AND ANALYZES

In MANET [2], achieving good Quality of Service (QoS) is a critical issue and is very difficult to guarantee mainly due to the dynamic nature of the network and the lossy nature of wireless links. In this context, routing protocols play a significant role. The main goal of any ad-hoc network routing protocol is to establish an efficient route between any two nodes with minimum routing overhead and bandwidth consumption. The protocols are different in terms of routing methodologies and the information used to make routing decisions [3]. The different routing approaches used in MANET can be classified into table driven protocols [4][5] and on-demand protocols [6][7][8]. Table driven protocols are proactive. In the proactive approach, the nodes maintain updated routing tables. To communicate, a path is immediately available for the source node. The drawback of this approach is that control messages are broadcasted periodically in the network. This leads to high routing overhead, limiting the network data communication

capabilities. On-demand approaches are source-initiated reactive mechanisms. With this routing approach, a path is issued at the request of the source node. In this study, we focus on reactive protocols. AODV is a well known protocol in this category. In recent years, much effort has been made to improve the standard AODV protocol [6]. In this section, after presenting the critical behaviors of the protocol, we review various proposed improvements. We conclude the section with a discussion on limitations of two important processes of AODV protocol.

A. AODV protocol

On-demand approaches are source-initiated reactive mechanisms. When a node needs to communicate and no route toward the given destination is available in its routing table, a route request packet is issued and flooded in the network [6].

Once the first RREQ packet reaches the destination node or an intermediate node with a fresh route toward the destination is reached, a route reply (RREP) packet is sent back to the source node. The source node rebroadcasts the RREQ if it does not receive a RREP during a Route Reply Wait Time (RREP_WAIT_TIME). It tries discovery of path up to a given maximum number of attempts and aborts the session if it fails. As the RREP packet is routed back along the reverse path, the intermediate nodes along the path record a tuple for the destination in their routing tables, which point to the node from which, the RREP is received. This tuple indicates the active forward route.

AODV uses a timer-based technique to remove stale routes promptly. Each routing entry is associated with a route expiration timeout. This timer is refreshed whenever a route is used. Periodically, newly expired routes are invalidated.

Route maintenance is done using route error (RERR) packets. When a link breakage is detected, routes to destinations that become unreachable are invalidated. RERR propagation mechanism ensures that all sources using the failed link receive the RERR packet. RERR packet is also generated when a node is unable to forward a data packet for route unavailability.

Broadcasted route request and route error messages may be important if established routes are much bits error-prone. This can be demonstrated by simulation with the use of a realistic physical layer and a realistic wave propagation model.

B. Enhanced AODV

The AODV protocol has two major problems: a long end-to-end communication delay due to overtime induced by route discovery process and an important routing load and communication delay variation when the frequency of link failures is high. Since the publication of standardized version of AODV, many efforts have been made to improve it. The major challenge is to limit the frequency of route discovery process. Thus, several optimizations have been proposed in the literature. Among them, we note taking into account link quality in the route selection process and adapting timers to the network dynamics.

1) *Tacking into account link quality*: In a wireless network, several factors impact on the quality of links. Among others, we can cite the distance between nodes, obstacles in the propagation medium, interference in the environment of communicating neighbors. Highlighting the impact of barriers in the propagation medium is only possible with a realistic propagation model. Taking into account link quality in the establishment of communication paths is an important factor for efficient use of network capacity. Correct estimation of link quality and choice of effective metrics are major problematics for QoS routing protocols designers. To take into account link quality in the route selection process, several methods are proposed with different QoS metrics including bandwidth, delay, packet delivery ratio, Bit Error Rate (BER). Fei et al. [9] present the design and selection of appropriate routing metrics as the principal issue to guarantee efficient routing in self-organizing networks. They attempt to analyze, compare and summarize traffic-based routing metrics in the Expected Number of Transmissions (ETX) family.

Khaled et al. [10] propose a path robustness-based quality of service routing for MANET. They proposed that before processing RREQ packet, an intermediate node must assure that its lifetime and the delay toward the neighbor from which, it receives the RREQ packet are above given delay-threshold and lifetime-threshold. At each hop, at least five checks are made and RREQ packet size increased with a node address. Destination node and source node must wait for copies (that have followed different paths) of RREQ and RREP packets until a timeout. The overhead (additionnal delay and routing load) and the complexity of this approach make problematic protocol effectiveness.

Some works, such as [11], use optimal link metric value in the path choice. Path selection choice based on optimal link metric value may not allow to get the best path. For example, for number of hops or retransmissions count-based metric, a path containing the link with the badest metric value m (compared to links of other feasible paths), is preferred to anyone, which links metrics values are upper than m , even if the other links quality of the first (path) are very good.

Some authors use additive and multiplicative metric to enhance AODV route discovery process. To find the optimal path in wireless mesh networks, Kim et al. [12] modify the standard AODV RREQ process. They propose that duplicate RREQs with better cumulative link metric value be forwarded, so that all the possible routes are considered. As link quality metric, they use an improved Expected Transmission Time (ETT) [13]. Their RREQ packet carries the cumulative link ETT value. They estimate the achievable throughput of their approach more than twice compared to standard AODV. We presume it is not necessary to re-broadcast duplicate RREQ packets. The intermediate node may note all possible reverse paths and retain as active reverse path to the source the better one according to the considered QoS metric. Their approach needs to be tested in MANET context with realistic simulation assumptions.

2) *Taking into account network dynamics*: Mobility of nodes is one of the essential issue of MANET. Taking into account the mobility of nodes is countered, first, by difficulties to adequately measure the mobility degree of a node. Many

papers [14][15] propose to privilege nodes with low speed but network topology change is not local problem. A node may be fixed but if its neighborhood moves a lot, integrating this node into transmission path will not allow efficient communication.

Some authors propose to use link breakage prediction for packet loss avoidance. In fact, when intermediate node detects degradation of neighbor link quality on active route, it may anticipate route maintenance process. Then, source node is advertized to the probable path failure and anticipates route recovery process. This avoids transmission interruption. QoS metrics used in this method include received signal strength [16], packet delivery ratio of control packets [17]. Very often, the power of modeled signal depends only on the distance to the concerned neighbor node. It is known that obstacles in wave propagation environment has an impact on signal strength [18][19]. Even if these metrics are accurately measured, the approach only anticipate the break of the link. The source must initiate a new route recovery process. The impact on delay improvement is not significant.

Amruta et al. [16] and Naif et al. [20] focused on accessibility prediction to restrict route discovery for future communications. Indeed, during the usual routing operations, a node can collect significant information enabling it to predict the accessibility and the relative mobility of the other nodes in the network. However, due to rapid change of network topology and since they are not actively maintained, these routes become obsolete. Macker et al. [21] study mobile routing path stability performance when constrained to use a distributed connected dominating set (CDS) control plane induced sub graph. They present weighted, degree-based Essential CDS (ECDS) results alongside those obtained using a CDS temporal stabilization algorithm. Compare to full topology shortest path forwarding (SPF) their work provides significant improvement in lifetimes for the ECDS election stabilization mode but at the price of additional average routing stretch.

Kirthana Akunuri [22] proposes a novel on-demand routing protocol, Speed-Aware Routing Protocol (SARP) to mitigate the effects of high node mobility by reducing the frequency of route disconnections in a MANET. SARP identifies a highly mobile node which forms an unstable link by predicting the link expiration time (LET) for a transmitter and receiver pair. Their work decreases control traffic but deteriorates other performance metrics like the throughput (i.e., number of packets received).

C. AOMDV

In classical AODV, a route discovery process allows the source node (initiator of the route request) to obtain a single path for its data transmission. It must re-initiate the process when the used path is broken. Since each route discovery induces high routing load and latency, frequency of use of this process should be kept low so that the routing is effective. Multipath routing protocols have been proposed to meet this objective.

Ad hoc On-demand Multi-path Distance Vector (AOMDV) [23] is a well known multi-path routing protocol. The key concept of this protocol is the computation and recording of multiple paths (to a given destination) by route

search. With these paths to a given destination, a node chooses a new route from backup routes when the active one (what was in use) is broken, thus avoiding having to re-initialize the route discovery process. A new route search process is only necessary when all the available routes fail. To form multiple paths, all duplicate RREQ packets received by a node are taken into account but not rebroadcast, as each RREQ defines an alternative route. Several studies have shown that the multipath approach improves AODV performance. However, in a context of very unstable network topology, AOMDV's performance are not stable.

Yufeng et al. [11] propose to improve the AOMDV protocol. They focus on the choice of path having a minimum number of retransmissions. However, a metric based on the number of retransmissions is not suitable in their context. Indeed, the route recovery process, the focal point for the effectiveness of reactive protocols, can not benefit from this improvement because the RREQ packet is not forwarded. Authors in [24] highlight a major drawback due to the fact that the relief routes are not maintained. Source node does not know if a given relief route is still valid when it is needed. The use of an obsolete path lead to increased average delay and jitter. They present two main contributions to improve the robustness of standard AOMDV protocol:

- the decentralized multi-path. The basic idea is to allow intermediate nodes to have multiple paths and locally repair broken routes,
- by a cross-layer approach, they take into account links reliability in the route choice process.

D. Analysis

In this sub-section, we present limits of the first received RREQ packet based route choice and issue of route maintenance especially in the case of multi-path routing. The *first RREQ* consideration approach means the selected path is the one with the better Round Trip Time (RTT). This path is the shortest one in term of hops count if all links are considered as similar. This path is the best path for control packets but not obviously the best for the data packets. Indeed, the RREQ packets, like for most of routing control packets, are small size packets. They are less vulnerable to interference. We study the impact of the packet size on the number of transmissions required to successfully communicate on a link and on bit error rate (BER). Simulation conditions are similar with those presented in presented in Section IV. These results show that the number of retransmissions required to successfully transmit on a link increases with the packet size (see Figure 1). Thus, a link that is very reliable for control packet transmission may require several retransmissions for successful data packet. It is not conceivable to use reasonably sized packet for route discovery (close to that of data packet) as this will cause constantly high routing load and much of the bandwidth is consumed by routing overhead.

Another questionable process of on-demand routing protocols is their route maintenance process. Contrary to proactive routing approaches, in on-demand routing methods, nodes maintain information only for active routes. In [24], it is shown that in mobility situation, AOMDV does not reach the expected

performance because the backup routes are obsolete when the active route is broken.

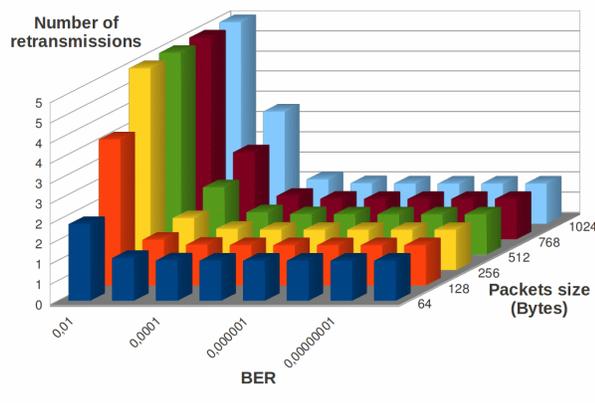


Figure 1. Number of expected retransmission according to packet size and BER link

III. QoS-BASED ON-DEMAND ROUTING PROTOCOLS

In this section, we present the PR-metric and three variants of AODV based on this metric. However, comprehensive presentation of this metric is beyond the scope of this paper.

A. QoS metric

The most commonly metrics used in QoS routing are bandwidth, delay, packet delivery ratio and bit error rate. Authors in [25][26][27][28][29] highlighted the problematic of link quality estimation and metrics design. An inaccurate design of link metric causes non-optimal route construction and thereby leads to end-to-end performance degradation. To partly solve this issue, retransmissions should be avoided whenever possible. BER is highly related to retransmissions. BER has a direct impact on Packet Delivery Ratio (PDR) and end-to-end delay, however, this metric has many drawbacks. Indeed, BER criteria is quite hard to measure in practice. A first method consists in injecting probe packets in the network. This method generates an additional load for the network [30]. Another approach consists in sending impulses and measuring the impulse response associated with a transmission. The main drawback is that this method requires an adapted physical layer. An estimation of all these disadvantages is presented in [25]. Moreover, using BER as an additive metric induces long end-to-end transmission path [31]. These long paths with an overall good BER value would potentially permit a better packet delivery ratio, but they generate long delay and induce poor throughput. Indeed, first, long paths increase intra-communication interference. Second, in mobility context, long path is very vulnerable.

For this study, we use a new metric based on the expected number of retransmissions required to communicate successful data packet on this link. Let us call it PR-metric. With PR-metric, distance between a node and its neighbor will not be 1 but $1 + a * (n - 1)$, where n represents the average number of transmissions required to make a data transmission

successful and a is a parameter to weigh retransmission cost. For retransmission, we want to design a transmission made after the first issue (after the first transmission attempt). The coefficient a is the ratio between the average time required for a retransmission over the time necessary for an initial successful transmission. Statistical analysis and results permit us to estimate a to 0.65 with 0.03 as standard deviation. Note that a is a mean value that represents retransmission cost. The experimental setup for this evaluation is similar to that is presented in Section IV. We investigated the simulated communications delays. Comparisons of delay where communication required a single transmission (one attempt) to those who need several retransmissions allowed us to obtain this value of a .

To evaluate this metric, we only have to get the number of packet transmissions. This information is available at the MAC level (it is a part of the communication statistics at the MAC layer) and, by a cross layer approach, is operated at routing level. We remind that there is no need to use special probes as in the estimation of most metrics. When the used packet size is small (like hello packet), the number of transmissions is almost always 1 (no retransmission). The large packages allow to better estimate the quality of a link with this metric. In our protocols all packets are taken into account. This metric has a direct impact on delay and throughput. Contrary to the well-known metrics like BER or ETX [32], it takes into account real time network load. Its estimation is local. It does not induce a significant routing load or a large computation time. It is a good compromise between the number of hops criterion and the BER or ETX criterion, which induces selection of long route [31].

B. QoS routing approach

QoS routing approach In a cross-layer approach, we use this criterion at the network layer. The objective is to avoid data transmission on bad paths in terms of BER. The following sub-sections present how we enhance routing protocols with BER information.

C. AODV-BL-PR

The basic idea of AODV-BL-PR is to remove from route establishment process poor quality links. Thus, the route search process will use a sub network with good quality links. AODV-BL-PR picks out AODV where we apply blacklisting approach to route recovery process. With AODV-BL-PR, when an intermediate node receives a RREQ packet, it compares the PR-metric value of link on which, this packet is received to a predetermined threshold. If this PR-metric value is higher than this threshold, the packet is discarded, otherwise it is managed as in standard AODV. Indeed, if this node forwards this request, it contributes to establish a bad path, which may cause high packet loss and high delay due to possible several retransmission attempts. Better paths may be found. We set this threshold to 2. We estimate that, in mobility context, after 2 attempts to transmit data, the path used is no longer valid. Note that maximum number of retransmissions at MAC layer is 4 for our test. We note that a control message (usually lighter) can be successfully transmitted on a poor quality link when a normal payload message can not be transmitted. With this route selection approach, paths containing bad links are disregarded.

This will also limit the dissemination of RREQ messages and then reduces routing overhead.

D. AODV-sum-PR

The basic principle of AODV-sum-PR during the route search process, is the choice of better route among the different possible routes between the source and the destination. To design this protocol, two main modifications are made to standard AODV, namely QoS-information dissemination and duplicate RREQ packets process by intermediate node.

- QoS-information dissemination: for AODV-sum-PR, RREQ and RREP packets are extended with the cumulative PR-metric (C-PR-metric) field. Source node initializes this metric to 0.0. An intermediate node increases the value of C-PR-metric by the PR-metric of the link on which, it received the packet. The intermediate node also integrates reverse path into its routing tables. Each entry is improved with the C-PR-metric as QoS-metric. The RREP packet also carries the C-PR-metric. The field is, this time, initialized to 0.0 by the destination node or to the current value of entry related to this destination by intermediate node, which initiates the RREP packet.
- Duplicate RREQ packet process: contrary to standard AODV, an intermediate node manages duplicate RREQ packet. Indeed, if the C-PR-metric of a duplicated RREQ packet is lower than the recorded one, the entry for source node (reverse path) is updated: the previous hop to the source node will be the new transmitter. Finally, the source node obtains a path to the destination with the lowest C-PR-metric value.

Note that intermediate node does not need to re-broadcast the duplicate RREQ packet and does not need to integrate the PR-metric value of all its neighbors as control packets header information, as widely done.

In Table I, we summarized the duplicate packet processing.

TABLE I. SAMPLE OF DUPLICATED PACKET PROCESSING ALGORITHM

for the concerned reverse path
if new C-PR-metric < current C-PR-metric
update next-hop
update C-PR-metric
else
drop the packet

E. AODV-new-timer

AODV protocol has several timers to manage the status of known routes and links with the neighborhood. These timers are updated through the various received packet management processes. The values of these timers are crucial for the protocol effectiveness, specially in the route announces by intermediate nodes. Indeed, an intermediate node, which knows a route to a desired destination, responds by RREP packet to the RREQ request. The announcement of an obsolete route leads, from the first attempt at data transmission, to the use of route repair process by a RERROR packet diffusion. In most different experiments, these parameters have the same

values in a static context and a dynamic context. However, in a context of mobility nodes, the neighborhood of a node changes quickly.

The basic idea of AODV-new-timer is to allow nodes to detect, as soon as, possible broken links or new links established. Then, in this enhanced protocol, we reduce the timers associated to the various recorded routes, established links with neighbors and waiting for a response (hello timer, route validity timer, waiting RREP packet timer, etc.). These timers are used to manage routes and links validation or recovery processes. To highlight the impact of these parameters and make a judicious choice of values, we have performed simulations and conduct a statistical study of the results of message exchanges. The simulation conditions are presented in Sub-section IV-B. The average speed of the 60 nodes used is 12m / s. 10 simultaneous end-to-end transmissions are initiated during 165s. We focused on the number of RERROR packets issued in the network. This performance parameter is used to highlight the level of network stability. The four sets of timers are presented in Table II. The obtained results are shown in Figure 2.

TABLE II. DIFFERENT SETS OF AODV TIMERS VALUES

Timer Parameter	Set1	Set2	Set3	Set4
MY_ROUTE_TIMEOUT	15s	10s	5s	2.5s
ACTIVE_ROUTE_TIMEOUT	15s	10s	5s	2.5s
REV_ROUTE_LIFE	10s	6s	3s	1.5s
BCAST_ID_SAVE	10s	6s	3s	1.5s
MAX_RREQ_TIMEOUT	15s	10s	5s	2.5s
RREP_WAIT_TIME	2s	1.0s	0.7s	0.4s
HELLO_INTERVAL	2s	1s	0.5s	0.3s
BAD_LINK_LIFETIME	6s	3s	1.5s	0.8s

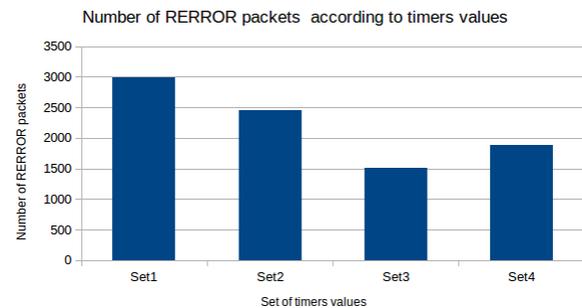


Figure 2. Number of RERROR packets according to the set of timers values.

These results have allowed us the choice of set 3 values (see Table II). It is these values that are used in the rest of the document for AODV-new-timer.

The new parameters are exhaustively presented in Table III. AODV-st means usual AODV. This coordinated reduction globally means that a node more frequently inventories its links and routes.

With this approach, we want to know the determining factor between taking into account link quality or a convenient control of neighborhood information for better performance in mobility context.

In summary, the reduction of route timeout value to 5s means that a path that is not used 5s ago is considered obsolete. The default value in standard AODV is 10s. The source waits

TABLE III. DEFAULT (AT LEFT) AND MODIFIED (AT RIGHT) AODV PARAMETERS FOR OUR TESTS

Timer Parameter	AODV-st	AODV-new-timer
MY_ROUTE_TIMEOUT	10s	5s
ACTIVE_ROUTE_TIMEOUT	10s	5s
REV_ROUTE_LIFE	6s	3s
BCAST_ID_SAVE	6s	3s
MAX_RREQ_TIMEOUT	10s	5s
NETWORK_DIAMETER	30 hops	10hops
RREP_WAIT_TIME	1.0s	0.7s
HELLO_INTERVAL	1s	0.5s
BAD_LINK_LIFETIME	3s	1.5s

less time (0.7 instead of 1.0) to restart a new request if it receives no response to a previous query. The network diameter is reduced to 10 instead of 30. We estimate that over 10 hops it is impossible to communicate in node mobility context. A HELLO_INTERVAL timer set to 0.5s instead of 1.0s, means that nodes should test their neighborhood more frequently.

IV. PERFORMANCE EVALUATION

In this section, we first present our simulation environment, we then present the results of simulation tests and analyze the performance of different protocols.

A. Simulation conditions

Simulation hypotheses have a major impact on the analysis of protocols' effectiveness. In wireless networks, the main factors that impact the probability of successful packet reception are the radio propagation environment, interference from other transmissions and link breakages due to dynamic topology of ad hoc network. On-demand and link states routing protocols suffer differently from these factors. To compare the efficiency of protocols, the effects of these factors must be taken into account. In this sub-section we want to highlight the impact of these simulation conditions.

Most research studies rely on simulation to show the effectiveness of their proposal. However, they do not take into account any environment when modeling propagation channel. They suppose that two nodes can communicate based on various empirical formulas. Often, the free-space model is used, only the direct ray between transmitter and receiver is considered and no obstacle disturbs transmissions. The two-ray-ground approach is also quite simplistic to compute such interferences. A realistic propagation model should consider path loss, fading and shadowing effects. If the environment is not considered, the obtained results can be biased and rather optimistic, since the influence of bad links is underestimated. In [18][19], authors show that interactions with wave propagation environment affect significantly link quality.

To better estimate the quality of links and to take into account the impact of propagation medium on the quality of communications, we use a realistic propagation simulator developed by XLIM-SIC laboratory [33] called Communication Ray Tracer (CRT). This software allows the modeling of the electromagnetic wave propagation in 3D environment. The paths between two points (transmitter and receiver) uses a 3D ray tracing technique. Thus, for a selected link, it determines the existing paths and their own characteristics like delay, attenuation, phase and polarization. These parameters depend

on the environment. It allows to fully characterize narrow and broad band channels and thus provides a realistic approach to multipath. With CRT, we conduct a semi-deterministic simulation of the entire chain of transmission (encoding, modulation, sound effects, demodulation and decoding). The impulse responses (IR) from the simulation are precisely computed and can thus be used, processed or analyzed easily afterward. From these impulse responses, they calculate the BER of the radio links by counting errors to the message originally sent.

Hamidouche et al. [18] highlight the overly optimistic results, when using free-space model propagation, compared to the ones provided by CRT. In these results, while the Free-space model provides a 100% as packet delivery rate, the CRT model offers about 60%. When the average number of hops is 1.2 in free-space, the CRT model offers a minimum of 2.5. To experiment quality of service, simulation should compute correct attenuation and error rate by taking into account not only shadowing and fading but also obstacles and multipaths effects.

Multi-communication effects must also be considered in a suitable way. Interference is an inherent property of wireless networks, which affects network efficiency as well as routing protocol performance [34]. As pointed out by Gupta and Kumar [35], the degradation of performance is observed when the number of nodes increases because each node has to share its radio channel with its neighborhood. Thus, in order to route data packets over non congested links and maximize overall network throughput, a protocol should focus on using available capacity of suitable links. On this subject, Jain et al. [36] advocate that routing or transport protocols in ad hoc networks should provide appropriate mechanisms to push the traffic further from the center of the network to less congested links. Some researchers emit important assertions about the correlation between number of nodes and source-destination throughput. Gupta and Kumar [35] prove that when the number of nodes n increases, the throughput per source-destination pair decreases approximately as $\mathcal{O}(1/n)$. Hekmat and Van Mieghem [37] reveal the existence of a network saturation point, after which, the network throughput no longer increases with respect to the number of nodes. Nevertheless, these assertions should be verified in realistic communication conditions.

In mobility situations, unrealistic mobility models (use of constant speed, pause time method, etc.) are very often used and interactions between mobile entities are not taken into account. It is shown in [38] that mobility model may drastically affect protocol performance.

B. Experimental setup

To compute more real simulations, we use a realistic wave propagation model taking into account environment characteristics. Therefore, we enhanced NS2 [39] with a ray-tracer simulator, Communication Ray Tracer (CRT) [33], that has been developed at the XLIM-SIC laboratory. CRT simulator provides a 3D ray-tracer wave propagation model. It takes into account the geographical data, electrical properties of materials, the polarization of the antennas, the position of the transmitters and receivers, the carrier frequency and the maximum number of interactions with the surrounding obstacles.

To realistically model node movement, we use the VANET-Mobisim [40] software. Node speed is computed by this software. The mobility model implemented is more realistic than widely used ones [41][42][38]. Paths are defined in correlation and consistency with our environment model. VANET-Mobisim is also easily interfaced with NS2. Specifically, VANET-Mobisim uses a mobility file in XML format, which contains all the detailed informations of the microscopic and macroscopic models that govern mobility of nodes. The mobility model used in this software takes into account the environmental parameters of the mobile nodes (traffic lights, speed limits, etc.) and possible interactions between mobile nodes. A node may thereby accelerate, decelerate according to environment constraints.

The global parameters for the simulations are given in Table IV.

TABLE IV. SIMULATION PARAMETERS.

Parameters	Values
Network simulator	ns-2
Simulation time	180s
Simulation area	1000m*1000m
Maximum number of transmissions	4
Transmission power	0.1w
Data types	CBR
Data packet size	512 bytes
MAC layer	IEEE 802.11a

We also use a realistic model of the Munich town (urban outdoor environment, see Figure 3), obstacles (building, etc.) are printed red. Dots represent nodes. Other real environments could be used in a more comprehensive study.

As routing protocols, we compare AODV-st, the standard AODV protocol [6], to the three enhanced ones presented in Section III.



Figure 3. Simulation environment when number of nodes=60. Obstacles are printed red.

C. Simulation results

In this section, we study the impact of mobility on performance of the four protocols. 60 mobile nodes move in the Munich town environment (Figure 3). Their average speeds range from 4m/s to 20m/s. 10 simultaneous end-to-end transmissions are initiated during 165s. As performance parameters we rely primarily on average end-to-end delay of data packets, PDR and Routing Overhead (RO). End-to-End Delay concerns only successfully delivered packets. PDR is the ratio of the number of successfully delivered data packets over the number of sent data packets. Routing overhead is the number of routing protocol control packets. It permits to evaluate the effective use of the wireless medium by data traffic.

In Figure 4, we present data transmission delay evolution according to the average node speed. These results show that AODV-new-timer outperforms the standard AODV and the two PR-metric based ones (AODV-BL-PR and AODV-sum-PR). The good performance of AODV-new-timer is explained by the fact differend waiting times are reduced and the near real time knowledge of neighborhood avoids node to process obsolete paths. Node implementing AODV-new-timer detects links breakage quickly. For QoS-based AODV (AODV-BL-PR and AODV-sum-PR), determining QoS routes requires substantial time and with node mobility, established routes become obsolete quickly. Thus, they are less efficient in delay parameter than AODV-st.

PDR evolution according to node speed is presented in Figure 5. Here again we see the same trends. The AODV-new-timer is better than the three other protocols. PDR values are higher than 58%. The paths established by the QoS protocols are longer than those established by the standard protocol. Their average path length is 3.4 against 2.6 for the standard protocol. In situations of mobility, long paths are more vulnerable. Also, over long paths, intra-interferences are more important as intermediate node may not receive a packet and retransmit another at the same time. These contribute to the poor performance in PDR and delay for these QoS protocols compared to the standard one. We also note that the number of packet loss on the first attempt of data transmission is 20% of total packet loss for AODV-st and PR-based ones against 13% for AODV-new-timer. This is explained by the use of obsolete path attempt by AODV-st and PR-metric based AODV.

Others criticism of blacklisting approach applied to AODV (AODV-BL-PR) may be made. An analysis of trace files shows that the percentage of transmission failures due to lack of route is high (10%) with AODV-BL-PR. This is explained by the fact the use of a threshold on the PR-metric to decide whether forward a Route Request packet may lead to ignore several alternative paths, thus reducing the reliability of the entire network. We remind that with this protocol, the links with PR-metric higher than 2 (equivalent to 3 transmission attempts) are excluded from the route search process. Finally, note that ignoring some bad links may induce a loss of network connectivity. In [43], it is emphasized that blacklisting policy could filter routing options severely, limiting the efficiency of the routing algorithm if an improper threshold is chosen.

A thorough analysis of the simulation shows that the majority of communications where source and destination

are far apart from each other have failed. Established routes become obsolete even before the first data packets arrive at the destination.

These show that better neighborhood information control is more important than taking into account link quality for AODV efficiency.

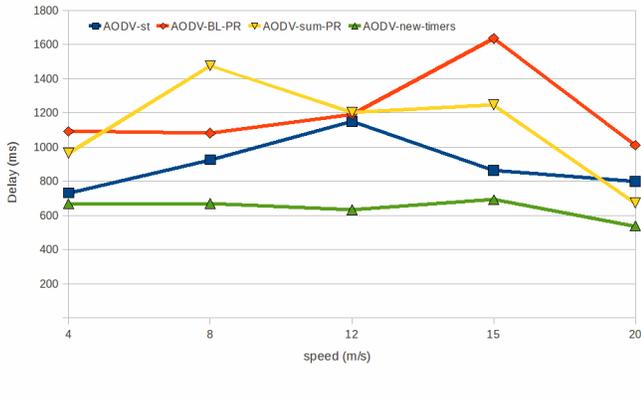


Figure 4. Delay evolution when speed increases.

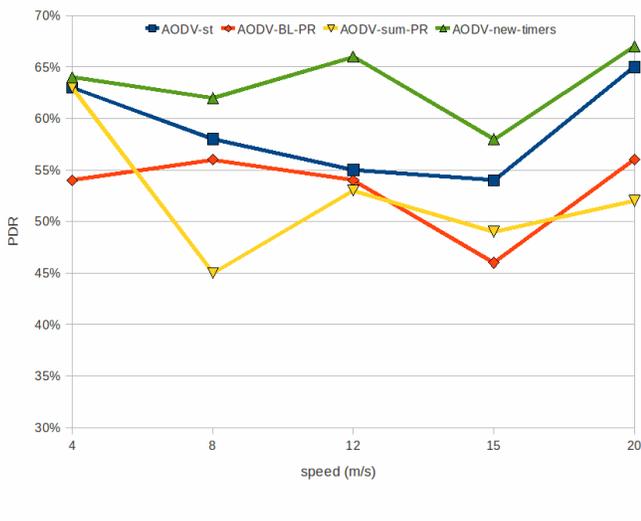


Figure 5. PDR evolution when speed increases.

Protocol's performance in RO parameter is presented in Figure 6. The high cost of AODV-new-timer is expected since Hello and RREQ messages emitting frequency increased. Its RO is, as expected, double that of AODV-st in a context of low mobility ($v = 8 \text{ m/s}$). This difference diminishes when the average node speed increases. This is explained by the fact that with AODV-new-timer, the recourse to route recovery process and therefore the dissemination of REROR packets is less. The better performance of QoS-based AODV compared to standard one can be explained by better paths selection. In addition, blacklisting approach of AODV-BL-PR limits the dissemination of RREQ messages. For AODV-new-timer, the

RO may be very high in a dense network and thus may impede the good performance of this protocol. A choice of timer values, depending on the volatility of links, could help ensure better compromise.

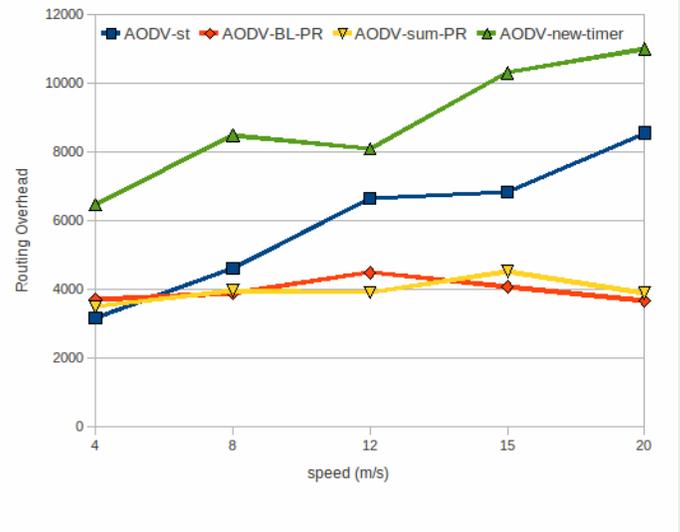


Figure 6. RO evolution when speed increases.

We also note that the curves performance are not monotonic. This denotes the complexity of the mastery of the network topology in mobility context. The jitter is high and raises problems for use of these networks in multimedia communications, especially voice communications.

V. CONCLUSION AND FUTURE WORK

The performance of AODV Protocol held in the efficiency of route discovery process. In an unstable links context, routing load and jitter may be significant. The main improvements are to limit the dissemination of route request messages and reduction of the frequency of route recovery process solicitations. However, in a context of mobility, link quality estimation procedures and better paths choose mechanisms must be efficient in terms of bandwidth and time consumption.

In our study, we tested the effectiveness of different QoS-based methods under realistic wave propagation model and realistic mobility model. For QoS metric, we use number of retransmissions count-based metric. Although we used a simple and effective method for link quality estimation, the results show that taking into account the quality of links is not effective for the MANET performances improvement. The additional complexity, induced by QoS management, increases delay and precipitated the obsolescence of the links.

To achieve better performance in high speed MANET context, the real challenge is the effective control of node neighborhood and accurate established routes lifetime and waiting RREP packet timeout value.

A more comprehensive study of the problematic of on-demand routing protocol performance could concern other real

environments (than Munich town one) and a refinement of the penalty coefficient due to retransmissions.

A solution where the inventory frequency of the neighborhood depends on the network dynamics might improve the performance of on-demand routing approach in mobility contexts.

ACKNOWLEDGEMENT

We would like to thank our colleagues of XLIM-SIC laboratory of University of Poitiers for simulation tools developed including Communication Ray Tracer (CRT) simulator.

We also thank the Polytechnic University of Bobo Dioulasso for facilitating this paper drafting.

REFERENCES

- [1] T. Yélémou and T. M. Dandjinou, "Problematic of QoS-based On-demand Routing Protocols to Improve Communication in Mobility Context," in *The Tenth International Conference on Wireless and Mobile Communications, ICWMC 2014, Seville, Spain*. IARIA, Jun. 2014, pp. 174–179.
- [2] I. Chlamtac, M. Conti, and J. J. N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13 – 64, 2003.
- [3] F. Sarkohaki and S. Jamali, "A survey of routing protocols for mobile ad-hoc networks with comprehensive study of olsr," *International Journal of Computer Science and Network Solutions*, vol. 2, pp. 1–26, March 2014. [Online]. Available: <http://www.techrepublic.com/resource-library/whitepapers/a-survey-of-routing-protocols-for-mobile-ad-hoc-networks-with-comprehensive-study-of-olsr/>
- [4] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF Network Working Group, Tech. Rep. RFC3626, Oct. 2003.
- [5] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *Proceedings of the 1st annual international conference on Mobile computing and networking*, ser. MobiCom '95. New York, NY, USA: ACM, 1995, pp. 86–95. [Online]. Available: <http://doi.acm.org/10.1145/215530.215560>
- [6] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing protocol," IETF Network Working Group, Tech. Rep. RFC3561, Jul. 2003.
- [7] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., 1996, vol. 353, pp. 153–181.
- [8] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *IEEE Computer and Communications Societies, Annual Joint Conference of the*, vol. 0, 1997.
- [9] F. Shi, D. Jin, and J. Song, "A survey of traffic-based routing metrics in family of expected transmission count for self-organizing networks," *Computers and Electrical Engineering*, vol. 40, no. 6, pp. 1801 – 1812, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045790614001700>
- [10] K. A. Al-Soufy and A. M. Abbas, "A path robustness-based quality of service routing for mobile ad hoc networks," in *2010 IEEE 4th International Conference on Internet Multimedia Services Architecture and Application (IMSAA)*. IEEE, Dec. 2010, pp. 1–6.
- [11] Y. Chen, Z. Xiang, W. Jian, and W. Jiang, "A Cross-Layer AOMDV routing protocol for V2V communication in urban VANET," in *5th International Conference on Mobile Ad-hoc and Sensor Networks, 2009. MSN '09*. IEEE, Dec. 2009, pp. 353–359.
- [12] S. Kim, O. Lee, S. Choi, and S. Lee, "Comparative analysis of link quality metrics and routing protocols for optimal route construction in wireless mesh networks," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1343–1358, Sep. 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870511000746>
- [13] P. Esposito et al., "Implementing the expected transmission time metric for OLSR wireless mesh networks," in *Wireless Days 2008 (WD08)*, Nov. 2008, pp. 1–5.
- [14] N. Enneya, K. Ouidi, and M. Elkoutbi, "Enhancing delay in manet using olsr protocol," *International Journal of Computer Science and Network Security IJCSNS*, vol. 9, no. 1, 2009.
- [15] M. Benzaid, P. Minet, and K. Al Agha, "Integrating fast mobility in the OLSR routing protocol," in *4th International Workshop on Mobile and Wireless Communications Network, 2002, 2002*, pp. 217 – 221.
- [16] A. Chintawar, M. Chatterjee, and A. Vidhate, "Performance analysis of ad-hoc on demand multipath distance vector routing protocol with accessibility and link breakage prediction," in *IJCA Proceedings on International Conference and workshop on Emerging Trends in Technology (ICWET)*, 2011, pp. 1–6.
- [17] A. Khosrozadeh, A. Akbari, M. Bagheri, and N. Beikmahdavi, "New AODV routing protocol with break avoidance," in *2011 International Symposium on Computer Science and Society (ISCCS)*. IEEE, Jul. 2011, pp. 303–306.
- [18] W. Hamidouche, R. Vauzelle, C. Olivier, Y. Pousset, and C. Perrine, "Impact of realistic mimo physical layer on video transmission over mobile ad hoc network," in *20th Personal, Indoor and Mobile Radio Communications Symposium, 2009*.
- [19] M. Guenes, M. Wenig, and A. Zimmermann, "Realistic mobility and propagation framework for manet simulations," in *NETWORKING 2007: Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet, Proceedings*, ser. Lecture Notes in Computer Science, vol. 4479. Springer Verlag, 2007, pp. 97–107.
- [20] N. Alsharabi, L. Ping, and W. Rajeh, "Avoid link breakage in on-demand ad-hoc network using packet's received time prediction," in *Proceedings 19th European Conference on Modelling and Simulation, 2005*.
- [21] J. Macker, D. Claypool, and N. Hughes, "Improving Routing Path Stability in Mobile Ad Hoc Networks That Use a CDS Control Plane," in *2014 IEEE Military Communications Conference (MILCOM)*, Oct. 2014, pp. 1099–1104.
- [22] K. Akunuri, "Design and analysis of a speed-aware routing protocol for mobile ad hoc networks," Ph.D. dissertation, Missouri University of Science and Technology, 2011.
- [23] M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Network Protocols, 2001. Ninth International Conference on*, Nov. 2001, pp. 14 – 23.
- [24] T. Yélémou, "New approach to improve the robustness of AOMDV protocol," in *2013 10th International Multi-Conference on Systems, Signals Devices (SSD)*, Mar. 2013, pp. 1–6.
- [25] L. Shi, A. Fapojuwo, N. Viberg, W. Hoople, and N. Chan, "Methods for calculating bandwidth, delay, and packet loss metrics in Multi-Hop IEEE802.11 ad hoc networks," in *IEEE Vehicular Technology Conference, 2008. VTC Spring 2008*. IEEE, May 2008, pp. 103–107.
- [26] M. Zuniga, I. Irzynska, J. Hauer, T. Voigt, C. A. Boano, and K. Roemer, "Link quality ranking: Getting the best out of unreliable links," in *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. IEEE, 2011, pp. 1–8.
- [27] D. Johnson and G. Hancke, "Comparison of two routing metrics in olsr on a grid based mesh network," *Ad Hoc Networks*, vol. 7, no. 2, pp. 374 – 387, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S157087050800053X>
- [28] S. Su, Y. Su, and J.-Y. Jung, "A novel QoS admission control for ad hoc networks," in *IEEE Wireless Communications and Networking Conference, 2007.WCNC 2007*. IEEE, Mar. 2007, pp. 4193–4197.
- [29] B. Bakhshi and S. Khorsandi, "Complexity and design of qos routing algorithms in wireless mesh networks," *Computer Communications*, vol. 34, no. 14, pp. 1722 – 1737, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0140366411001198>
- [30] J. Ledy, H. Boeglen, A. Poussard, B. Hilt, and R. Vauzelle, "A Semi-Deterministic channel model for VANETs simulations," *International Journal of Vehicular Technology*, vol. 2012, pp. 1–8, 2012. [Online]. Available: <http://www.hindawi.com/journals/ijvt/2012/492105/>
- [31] T. Yélémou, P. Meseure, and A.-M. Poussard, "Improving zrp performance by taking into account quality of links," in *WCNC 2012 Paris, France, April 2012*.

- [32] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "a high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, pp. 419–434, Jul. 2005. [Online]. Available: <http://www.springerlink.com/content/x11047934qu8h56u/>
- [33] R. Delahaye, "Efficient and realistic simulation of the physical layer to aid routing in ad hoc networks." Ph.D. dissertation, University of Poitiers, 2007.
- [34] G. Parissidis, "Interference-aware routing in wireless multihop networks," Ph.D. dissertation, University Paris VI, 2008.
- [35] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [36] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "The impact of interference on multi-hop wireless network performance," in *2nd ACM MOBICOM 2003 Conference, San Diego, CA, Sep 19, 2003*, vol. 11, 2005, pp. 471–487.
- [37] R. Hekmat and P. Van Mieghem, "Interference in wireless multi-hop ad-hoc networks and its effect on network capacity," *Journal on Wireless Networks (Proceedings of Conference on Ad hoc networking)*, vol. 10, no. 4, pp. 389–399, Aug. 2004.
- [38] F. Bai, N. Sadagopan, and A. Helmy, "Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *22th Annual Joint Conference of the IEEE Computer and Communications*, vol. 2, 2003, pp. 825–835.
- [39] L. Breslau et al., "Advances in network simulation," *COMPUTER*, vol. 33, no. 5, May 2000.
- [40] J. Haerri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks : a survey and taxonomy," in *IEEE communications surveys and tutorials*, vol. 11. IEEE-INST Electrical Electronics Engineers INC, 2009, pp. 19–41.
- [41] Q. Zheng, X. Hong, and S. Ray, "Recent advances in mobility modeling for mobile ad hoc network research," in *Proceedings of the 42nd annual Southeast regional conference*. ACM, 2004, pp. 70–75.
- [42] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '99. New York, NY, USA: ACM, 1999, p. 5360. [Online]. Available: <http://doi.acm.org/10.1145/313237.313248>
- [43] O. Gnawali, M. D. Yarvis, J. Heidemann, and R. Govindan, "Interaction of retransmission, blacklisting, and routing metrics for reliability in sensor network routing," in *IEEE Conference on Sensor and Adhoc Communication and Networks*, pp. 34-43. Santa Clara, California, USA, IEEE., May 2004. [Online]. Available: <http://www.escholarship.org/uc/item/0nw798m9>

HQMR: Hybrid QoS based Routing Protocol for Wireless Mesh Environment

Hajer Bargaoui, Nader Mbarek, Olivier Togni

LE2I Laboratory, University of Burgundy
Dijon, France

e-mails : {Hajer.Bargaoui, Nader.Mbarek,
Olivier.Togni}@u-bourgogne.fr

Mounir Frikha

MEDIATRON Laboratory, High School of Communication
of Tunis (SUP'COM)

Tunis, Tunisia

e-mail : m.frikha@supcom.rnu.tn

Abstract—Wireless Mesh Networks (WMNs) have been attracting more and more interest from both academic and industrial environments for their seamless broadband connectivity to Internet networks. Besides, providing QoS guarantees for real-time and streaming applications such as Voice over IP (VoIP) and Video on Demand (VoD) is a challenging issue in such environment. Thus, we propose a novel QoS based routing protocol for wireless mesh infrastructure, called Hybrid QoS Mesh Routing (HQMR). Moreover, a clustering algorithm is developed to enhance scalability issues within the mesh infrastructure. HQMR is composed of two routing sub-protocols: a reactive routing protocol for intra-infrastructure communications and a proactive QoS based multi-tree routing protocol for communications with external networks. The proposed routing protocol ensures forwarding real-time and streaming applications with QoS guarantee in a mesh wireless environment. We analyze in this paper the simulation results of different scenarios conducted on the network simulator ns-3 to demonstrate the effectiveness of the reactive routing sub-protocol in forwarding real-time applications with QoS guarantee.

Keywords-Wireless Mesh Network; QoS routing; HQMR; Network Simulator; ns-3.

I. INTRODUCTION

Recently, wireless mesh networks have received increased attention from researchers and industrial environments [1]. They have emerged as a key wireless technology for numerous applications such as broadband home networking, community and neighborhood networks, enterprise networking, etc. [2], [3]. Besides, they are a promising solution to provide last-mile connectivity to Internet for fixed and/or mobile users in zones where wired networks deployment is difficult. These abilities are provided thanks to their various qualities. They are self-organizing and self-configuring networks where participating nodes automatically establish and maintain connectivity. They enable also quick deployment, easy maintenance, low cost, high scalability, etc. These benefits have motivated consistently researchers to study their characteristics for better performance.

In fact, a wireless mesh network is composed of two types of node: wireless mesh routers and wired/wireless mesh clients. The mesh routers are static and non-power constrained nodes and the mesh clients are potentially mobile nodes. In fact, mesh routers communicate between each

other in multi-hop fashion, forming a relatively stable network and the mesh clients are connected to these routers using a wireless or a wired link. The role of most mesh routers in a wireless mesh network is to perform relaying of data for other mesh routers, a typical ad-hoc networking paradigm. Some other mesh routers have also additional gateway capabilities. These nodes, named mesh gateways, enable the integration of wireless mesh networks with various other networks and often have a wired link to Internet, helping in forwarding clients traffic and in providing Internet services to the mesh clients.

One major challenge for wireless mesh networks is to provide QoS support. Since deployments of WMNs continue to grow, providing Quality of Service for real-time and streaming applications, such as VoIP and VoD, is an important task. Moreover, establishing paths with the highest performance is a challenging issue for routing protocols within wireless mesh networks in order to satisfy applications' requirements.

However, the different research works proposing routing solutions on wireless mesh networks rely simply on adapting protocols originally designed for mobile ad hoc networks and adding a little support for QoS. In this paper, we propose a hybrid QoS based routing protocol, called Hybrid QoS Mesh Routing (HQMR) [1], which exploits more efficiently the particular topology of a wireless mesh network, based on a hybrid wireless mesh architecture. The proposed wireless mesh architecture is formed by an IEEE 802.16j based infrastructure and different IEEE 802.11s based client domains. Furthermore, in order to solve scalability issues and reduce efficiently the network's load, a clustering algorithm is proposed for the IEEE 802.16j infrastructure of our global wireless mesh architecture. HQMR is then deployed on the IEEE 802.16j infrastructure to ensure routing functionalities. It is a hybrid protocol adopting a reactive routing sub-protocol for intra-infrastructure communications and a proactive multipath tree-based routing sub-protocol for inter-infrastructure communications, where the mesh gateway is considered as a root.

The remainder of this paper is organized as follows. In Section II, we present two standards of wireless mesh networks. Related works are presented in Section III. Section IV introduces the architecture of our framework. Then, the proposed HQMR routing protocol is defined in Section V. Section VI defines two usage scenarios of HQMR to illustrate its processing. We introduce respectively, the performance evaluation of IMRR routing sub-protocol and

the results analysis in Section VII and Section VIII. Finally, Section IX concludes the paper.

II. STATE OF THE ART

Given the increased interest in wireless mesh networks, different standards have been specified. In this section, we present the latest standardization results namely IEEE 802.11s standard based on Wi-Fi technology and IEEE 802.16j standard built on the WiMAX technology.

A. IEEE 802.11s Standard

The IEEE 802.11s standard started initially as a study group in 2003, and became a Task Group in July 2004 for developing a flexible and extensible solution for wireless mesh networks based on IEEE 802.11 technology. The first draft was accepted in March 2006 and their work was approved by 2011 [4].

An IEEE 802.11s network is formed by a wireless infrastructure, composed of a set of mesh routers named Mesh Points (MP), to which the mesh clients (STA) are connected to access the Internet services. Some MPs, named Mesh Access Point (MAP), have additional access point functionalities to help connecting the STA nodes to the mesh infrastructure. Other MPs named Mesh Portal Point (MPP) have gateway functionalities to ensure the connection between the mesh cloud and the external network.

Besides, IEEE 802.11s standard defines a layer 2 basic routing protocol named HWMP (Hybrid Wireless Mesh Protocol) and, therefore, uses MAC addresses and a radio routing metric.

B. IEEE 802.16j Standard

IEEE 802.16j task group was officially established in March 2006 and their work was published in 2009. The IEEE 802.16j standard [5], is an amendment to the IEEE 802.16e [6] standard in order to introduce Mobile Multi-hop Relay (MMR) specifications where traffic between a Multi-Relay Base Station (MR-BS) and a Subscriber Station (SS) can be relayed through nodes named Relay Stations (RS). The number of hops between MR-BS and SS is not defined but it must only contain RS nodes.

In fact, IEEE 802.16j has defined two different relay modes: transparent mode and non-transparent mode. In transparent mode, the RS is used to improve the network capacity. It does not forward any signaling frame. It relays only data traffic, that is why the SS, which is physically attached to it, does not know the existence of the RS. The non-transparent mode is usually used to extend the network coverage. The RS nodes in this mode are able to generate their own signaling frame or forward those provided by the MR-BS depending on the scheduling mechanism.

Just like the previous Wimax standard namely IEEE 802.16e [6], IEEE 802.16j manages also the quality of service at the MAC sub-layer by differentiating five service classes, from high to low priority: Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), enhanced real-time Polling Service (ertPS), non real-time Polling Service (nrtPS) and Best Effort (BE).

III. RELATED WORK

A. QoS Routing

QoS provisioning is an important issue for wireless mesh networks since they are typically used for providing broadband wireless Internet access to a large number of users and networks. To meet applications' QoS requirements, different QoS routing protocols were proposed for WMNs.

Wireless Mesh Routing (WMR) [7] is a QoS solution for wireless mesh LAN networks. It provides QoS guarantees in terms of minimum bandwidth and maximum end-to-end delay. These two parameters are verified jointly with the route discovery process. The value of the node's available bandwidth is estimated thanks to the bandwidth already in use by the considered node and by its neighboring nodes. Then, the end-to-end delay is estimated by using the round trip delay method [8]. Kon et al. [9] improve the WMR protocol by proposing a novel end-to-end packet delay estimation mechanism with a stability-aware routing policy. The delay estimation is based on packets named DUMMY-RREP, which have the same size, priority and data rate as real data traffic. The robustness of a link is estimated by measuring the number of Hello packets received during a given time.

Some other works include the QoS verification in the route discovery phase. For example, QoS AODV (QAODV) [10] integrates a new metric for IEEE 802.11 mesh networks, composed of bandwidth, delay, hop count and load ratio. In the same way, Rate-Aware AODV (R-AODV) [11] uses minimum network layer transmission time as a performance metric in multi-rate WiFi mesh networks. Mesh Admission control and QoS Routing with Interference Awareness (MARIA) [12] is another QoS aware routing protocol for wireless mesh networks. It is a reactive protocol incorporating an interference model in the route discovery process. This protocol uses a conflict graph model to characterize both inter and intra-flow interference. The available residual bandwidth is computed based on the maximal clique constraints in its local conflict graph to make distributed hop-by-hop admission control decision.

In this context, we propose the HQMR protocol to provide QoS provisioning routing functionalities within a wireless mesh environment.

B. Clustering

Clustering concept was introduced to organize large wireless multi-hop networks into groups named clusters. Every cluster is coordinated by a cluster-head to achieve basic network performances, even with mobility and limited energy resources. The different clustering algorithms differ mainly in the method used for the election of the cluster-heads: Lowest-ID heuristic [13], Highest-degree heuristic [14] and node-Weight heuristic [15]. The Lowest-ID algorithm [13] designs the node with the lowest-ID as cluster-head. Then, a cluster is formed by that node and all its neighbors. In order to maintain inter-clusters connectivity, Gateway-nodes are defined. The Highest-degree algorithm [14] uses the degree of the node (number of the neighbors) for cluster-head election process. The third type of clustering

algorithms calculates a weight for each node according to specific metrics. For example, the authors in [15] propose an algorithm that takes into consideration the number of nodes a cluster-head can handle ideally without any severe degradation in the network performances, transmission power, mobility and battery power of the node.

Combining clustering algorithms with routing protocols offers better performances within the network layer, by reducing the amount of control messages propagated inside the network since the exchange is limited within a cluster; and by minimizing the size of routing tables at each node since it stores only the information of its cluster.

Zone Routing Protocol (ZRP) [16] is a cluster-based routing protocol for ad hoc networks that uses different routing sub-protocols for inter and intra-clusters communications. Within a cluster zone, a proactive component is used to maintain up-to-date routing tables. Routes outside the routing zone are explored with a reactive component combined with a border-casting concept. This concept utilizes the topology information provided by the proactive protocol to direct query requests to the nodes in the border of the zone.

Singh et al. [17] propose a hierarchical cluster based routing protocol for wireless mesh networks, in which the mesh gateway is the highest level node. When a node has a data packet to forward, it sends a path request message to its cluster-head. In case that the destination is not in the same group, the cluster-head sends the path request message to the mesh gateway, which forwards the request to the other cluster-heads. Similarly, the research work in [18] defines a multi-level clustering approach with a reactive routing protocol for wireless mesh networks, in order to reduce the load on the mesh gateway. The source node unicasts the route request message to its cluster-head. If a route is not found, the cluster-head forwards the message to an upper-level node, firstly to a Group Head and then to the mesh gateway. This approach reduces considerably the number of broadcast messages used for route discovery process.

For its benefits, we adopt this concept while adapting it to cluster based routing for our HQMR protocol to solve scalability issues and to offer better routing performances within the wireless mesh infrastructure.

C. Multipath Routing

Multipath routing is the technique of using multiple paths between each node pair instead of having a single path, which helps to improve the available bandwidth, to reduce end-to-end delay and to enhance load balancing and fault tolerance [19]. These multiple paths may be used as backup paths or as concurrent paths. The backup paths are used only when the primary path is broken. The concurrent paths are used simultaneously to forward traffic, according to specific traffic distribution mechanism over the used paths [19], [20].

Providing broadband wireless Internet access to end users is an important objective of wireless mesh networks. Thus, most of the traffic is directed either from or towards the Internet mesh gateway. Consequently, some nodes or links could be overloaded since each node will aim to choose the best path to the gateway. The Multi-path Mesh (MMESH)

[21] protocol was proposed as a possible solution. It is a proactive multi-path routing protocol, specifying an algorithm to split traffic over multiple selected paths between each node and the mesh gateway, for balancing network load uniformly. Then, MMESH applies a congestion aware approach to choose the best path. Multi-path routing protocols help to improve the performances of a network by using multiple disjoint paths. However, when all these paths are utilized simultaneously to transmit data, they will affect each other by causing route-coupling problem [22]. AODV Decoupled Multipath (AODV-DM) [23] was developed to establish efficiently node-disjoint paths that are enough separated to avoid inter-path interferences. It selects the primary path according to the single path selection process. Then, a region is defined around this primary path, so that the second path will be selected outside it. Thereby, these two paths are not only disjoint, but also decoupled paths.

Multi-path Hybrid Routing Protocol (MHRP) [24] is a multipath routing protocol used with a hybrid architecture formed by a mesh infrastructure and ad hoc client domains. This protocol is based on the backup routes concept to enhance the network performance.

Considering the importance of the traffic from/towards the mesh gateway in a wireless mesh network, we propose the integration of the multipath routing concept as a promising solution for network load balancing in our wireless mesh infrastructure.

IV. PROPOSED GLOBAL HYBRID WIRELESS MESH ARCHITECTURE

For our framework, we adopt a hybrid wireless mesh network architecture, combining two different technologies. It is formed by a non-transparent IEEE 802.16j-based infrastructure and IEEE 802.11s-based client domains (Fig. 1). A hybrid QoS based routing protocol (HQMR) is also proposed within the wireless mesh infrastructure.

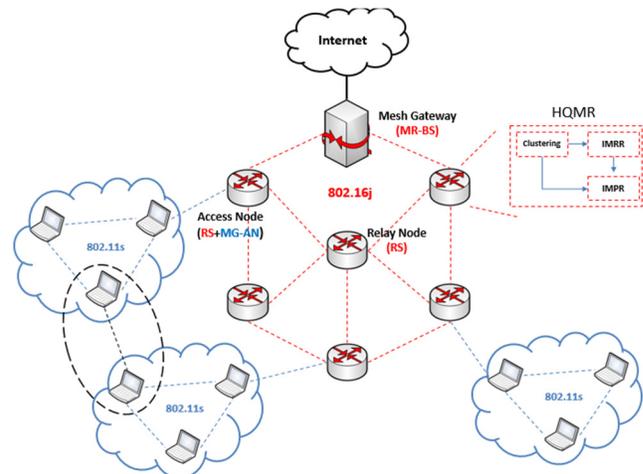


Figure 1. Global hybrid wireless mesh architecture

A. IEEE 802.16j-based mesh infrastructure domain

For the wireless mesh infrastructure, we use the non-transparent relay mode of the IEEE 802.16j technology to

ensure a better coverage. Then, in order to organize the functionalities of each node, we define three types of nodes within the mesh infrastructure: the Mesh Gateway (MG), the Relay Nodes (RN) and the Access Nodes (AN). The MG is the intermediate node between the Internet cloud and the wireless mesh infrastructure. It helps forwarding clients requests to the Internet network. The RNs are the nodes located in the core of the mesh infrastructure to ensure forwarding traffic flows from a node to another inside it. Last, we consider the nodes located in the border of the infrastructure, as Access Nodes (AN). They provide interconnection between the mesh infrastructure and the client domains. Thus, compared to the topology of an IEEE 802.16j network, our MG and RN nodes have, respectively, the same functionalities as the MR-BS node and the RS nodes. In fact, the AN nodes may be considered as bridge nodes playing both the role of a relay node in the IEEE 802.16j infrastructure and the role of a gateway in the IEEE 802.11s area. Thus, they are equipped with two radio interfaces: one is operating with the WiMAX technology [6] and another with Wi-Fi technology [25].

At each relay node of the wireless mesh infrastructure (including the ANs and the MG), the proposed routing protocol (HQMR) must be implemented with our clustering algorithm to reduce mainly the size of the routing tables. The different blocks of HQMR will be described in Section IV.

B. IEEE 802.11s-based mesh client domain

The client domains are formed by a set of IEEE 802.11s [4] MP (Mesh Point), which are interconnected to each other forming the mesh topology and by a gateway node that we called Mesh-Gateway Access Node (MG-AN). The MG-ANs have the functionality of the 802.11s MPP (Mesh Portal Point) implemented in the access node (AN) of our mesh infrastructure. So, in order to connect to the Internet cloud, the mesh clients forward, first, their traffic to their own gateway (i.e., MG-AN), for accessing the mesh infrastructure. Then, the MG-AN forwards directly the received traffic from its mesh clients to its own gateway.

V. HYBRID QoS MESH ROUTING

HQMR, our proposed protocol, is used to ensure routing functionalities within the wireless mesh infrastructure of our global wireless mesh architecture. It is a hybrid QoS-based routing protocol composed of two different routing blocks. The first routing sub-protocol Intra-Mesh infrastructure Reactive Routing (IMRR) is designed to forward communications within the infrastructure in a reactive manner, while the second routing block Inter-Mesh infrastructure Proactive Routing (IMPR) is deployed to forward communications to the external networks, particularly to the Internet network. The second routing sub-protocol is a tree-based multipath routing protocol, with the Mesh Gateway as a root of the routing tree.

Moreover, in order to improve the performance of our routing protocol, we adopt the concept of clustering to divide the topology of the infrastructure into a set of groups called clusters, each coordinated by a cluster-head. This division allows the network to minimize effectively the load of the

control messages since the exchange would be limited to a cluster domain. It helps also in reducing the size of the routing table at each node and simplifies routes discovery process thanks to the inter-clusters communications approach. Besides, this concept of clustering is considered as the most suitable solution to ensure the network scalability.

In this section, we present the algorithm specified for the clusters elaboration within the wireless mesh infrastructure and we introduce the two routing sub-protocols of HQMR. Before that, we define the mechanism used to provide the needed information about each node's neighbors and we specify the different QoS parameters and their estimation method to guaranty the QoS based routing characteristic of our proposed HQMR protocol.

A. Neighborhood Maintenance

Neighborhood information is very important for our protocol in order to provide the local topology (node's different neighbors), the necessary information for our clustering algorithm and the available QoS toward each neighbor. To maintain this information, every node in the network is required to send out periodically a Hello message (Table I), announcing its existence and its cluster information such as its state in the cluster, its calculated weight parameter used for cluster-head election, its CH's IP address (ID-CH) and its used bandwidth parameter. By receiving the Hello message from the different neighbors, each node updates its Neighbor Table (Table II), which is used to store for each neighbor its IP address (ID), all the needed information for clusters formation (Weight, State, ID-CH) and the available QoS parameters, including the available bandwidth, the delay and the jitter parameters.

TABLE I. HELLO MESSAGE FORMAT

ID	Weight	State	ID-CH	Used Bandwidth
----	--------	-------	-------	----------------

TABLE II. NEIGHBOR TABLE (NT)

ID	Weight	State	ID-CH	QoS Metric
----	--------	-------	-------	------------

B. QoS Routing Metrics

The purpose of our routing protocol is to find paths, which can satisfy the QoS requirements of real-time flows. The set of QoS requirements includes the bandwidth, the delay and the jitter parameters.

1) Available Bandwidth metric

To estimate the available bandwidth, each node considers the used bandwidth by its flows and the consumption of its neighbors announced in the Hello messages (1).

$$B(v) = B - \sum_{v' \in N(v)} B_{\text{used}}(v') \quad (1)$$

where $B(v)$ is the estimated available bandwidth by a node v , B is the total Bandwidth, B_{used} is the bandwidth used by a node and $N(v)$ in the neighborhood of the node v .

Then, the bandwidth parameter of the entire path is determined as the minimum bandwidth estimated at each node toward the destination.

2) Delay Metric

This metric estimation is based on measuring the round trip delay time (RTT) [8] of the Hello messages, which represents the time between initiating a Hello message and receiving a response. The delay of a path is the sum of its links delay metric.

3) Jitter Metric

The jitter metric defines the delay metric variation. It is estimated by calculating the mean of the differences between the RTT values for a specific period. Besides, the Jitter of a path is calculated by summing the Jitter of each link.

C. Clusters formation algorithm

Our clustering algorithm is a variant of the LID-based clustering algorithm [13] combined with the use of the weight concept developed by the Weighted Clustering Algorithm (WCA) [15] for the election of cluster-heads. Thus, a cluster is formed by the node with the lowest weight and all its neighbors. The same procedure is repeated among the remaining nodes, until each node is assigned to a cluster. Inter-clusters connectivity is maintained by defining some Gateway-nodes (Sub-Section 3), named Cluster Gateway (C-Gw) and Distributed Gateway (D-Gw). Moreover, in our adapted algorithm, we have opted for one-hop clusters to reduce the load of control messages within a cluster and to ensure a line of sight between the different cluster-heads and gateway nodes, which is an important characteristic for the deployment of our second routing sub-protocol IMPR (Section E). An example of a clustered wireless mesh infrastructure is illustrated in Fig. 2.

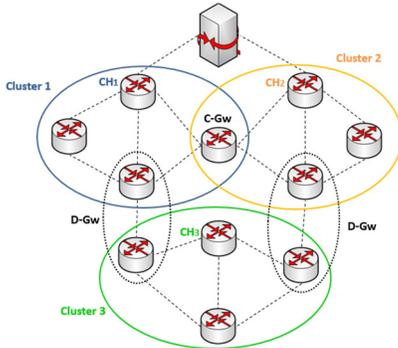


Figure 2. Clustered architecture of the wireless mesh infrastructure

Our clustering algorithm is composed of three main functions, which are presented in the following sub-sections: weight calculation, cluster-head election and clusters elaboration process.

1) Weight Calculation

In our algorithm, the weight assigned to each node is based on the WCA algorithm [15]. The latter takes into account the degree (neighbors' number), the transmission power, the mobility and the battery power of each node. It optimizes the degree of each cluster-head by choosing an optimal number M of nodes per cluster (M is a pre-defined threshold). This restriction aims that the cluster-head would be able to support ideally the nodes within its cluster.

However, given the stability of the nodes within our wireless mesh infrastructure, we are only interested in the first two parameters used to calculate the weight of WCA to find the optimal number of nodes within the transmission range and to estimate the transmission power toward the neighbors of a node. In addition, since most of the traffic is oriented to the Mesh Gateway, a third parameter is used in our weight calculation to take into account the power transmission of the node toward the Mesh Gateway. By this way, the cluster-head will be elected among the nearest nodes to the Mesh Gateway. Thus, the weight is calculated according to (2)-(6):

$$W_v = a \cdot \Delta v + b \cdot D_v + c \cdot DP_v \quad (2)$$

where a , b and c are the weighing factors so that $a+b+c=1$ and W_v is the weight of a node v .

$$d_v = |N(v)| = \left| \left\{ v' \in V, v' \neq v, \text{dist}(v, v') < tx_{range} \right\} \right| \quad (3)$$

where V is the neighborhood of a node v .

$$\Delta v = |d_v - M| \quad (4)$$

$$D_v = \sum_{v' \in N(v)} \text{dist}(v, v') \quad (5)$$

$$DP_v = \text{dist}(v, MG) \quad (6)$$

Equation (4) represents the degree-difference for a node v to compare its number of neighbors (3) to the optimal number of nodes that a CH may coordinate efficiently. The transmission power toward the neighbors is estimated in (5) by computing the sum of the distances with all its neighbors. Specially, the third parameter namely the transmission power toward the Mesh Gateway is calculated in (6).

2) Cluster-head Election

Initially, all the nodes are in the initial state that is the "Undecided" state and with a weight equal to zero. Thanks to the periodic exchange of Hello messages, the Neighbor Table (Table II) will be updated with the last calculated value of weight (W) for each neighbor. Each node waits for a period T_e before starting the selection of the cluster-heads, so that all the nodes have updated their NT (Neighbor Table). After this period, the node with the lowest W among its neighbors changes its state to "CH" and broadcasts a Hello message, as illustrated in Fig. 3.

```

1: If  $W_i = \min(NT[\text{weight}])$  then
2:    $S_i = CH$ 
3:    $ID\_CH_i = ID_i$ 
4:   Broadcasts Hello ( $ID_i, W_i, S_i, ID\_CH_i, B_{used}$ )
5: End If

```

Figure 3. Cluster-head election algorithm for node i

3) Clusters elaboration process

The division of the network into a set of clusters is based on the exchange of Hello Messages between each node and its neighbors. Fig. 4 illustrates the algorithm of the clusters elaboration.

```

On receiving a Hello message:

1: If (Hello [State] = CH) then {
2:   If ID-CHi = null then {
3:     Si = CM
4:     ID-CHi = Hello [ID-CH]
5:     Update (NT)
6:     Broadcast Hello (IDi, Wi, Si, ID-CHi, Bused)
7:   } Else {
8:     If (Hello [ID] < ID-CHi) then
9:       G = ID-CHi
10:      ID-CHi = Hello [ID]
11:    } End If
12:    Si = C-Gw
13:    Update (GwT)
14:    Broadcast GW-D (IDi, Wi, C-Gw, ID-CHi, G, null)
15:  } End If
16: Else if (Hello [State] = CM) then {
17:   If (Si = CM) then {
18:    If (ID-CHi = Hello [ID-CH]) then
19:      Update (NT)
20:    } End If
21:    Si = D-Gw
22:    Update (GwT)
23:    Broadcast GW-D (IDi, Wi, D-Gw, ID-CHi, Hello [ID-CH], Hello [ID])
24:  } Else
25:    Update (NT)
26:  } End If
27: Else
28:   Update (NT)
29: End If
30: If (ID-CHi = Hello [ID-CH]) then Update (NT)
31: Else {
32:   Update (NT)
33:   Update (NCHT)
34: }
35: End If
36: End If
    
```

Figure 4. Clusters elaboration algorithm

According to our algorithm, we distinguish five possible states of a node within a cluster. Besides, it is important to notice that the clustering algorithm is executed on each node of the infrastructure except the Mesh Gateway. The latter has its own state MG as Mesh Gateway. For the rest of nodes, we have the following states:

- **Undecided:** it is the initial state indicating that the node does not yet belong to any cluster.
- **Cluster Member (CM):** it is a node, which belongs already to a cluster. It changes its state from Undecided to CM, once it has received a Hello message from a CH.
- **Cluster Head (CH):** it is the node with the lowest weight and it is responsible of its cluster management.
- **Cluster Gateway (C-Gw):** it is a node in direct vision with two different cluster heads at the same time. It acts as a bridge between the two clusters. In fact, this node exists when two cluster-heads are at two-hops of each other.
- **Distributed Gateway (D-Gw):** it is a CM that has a neighbor belonging to another cluster. D-Gw ensures the communication between two disjoint clusters. This is the

case where two cluster-heads are at 3-hops of each other.

These different states with the different necessary transition conditions are described in a FSM diagram (Fig. 5).

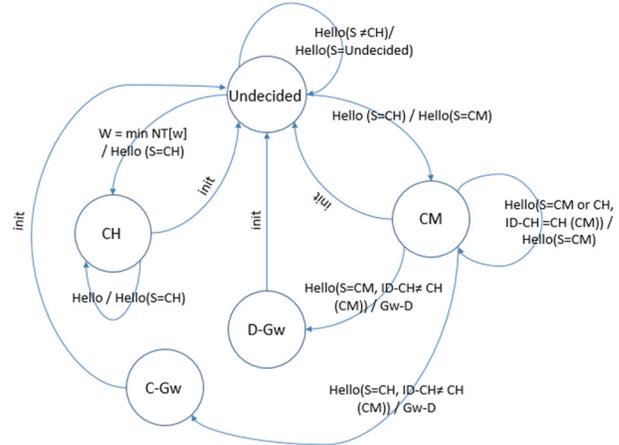


Figure 5. FSM of a node participating in the clustering algorithm

A node becomes a CM node when it receives a Hello message for the first time from a CH node. This node may change its state to a gateway node to ensure interconnection between two clusters. It may become a C-Gw when receiving a Hello message from another cluster-head. It changes its state to a C-GW and updates its Gateway Table (Table III), in which it keeps its type as gateway and the two interconnected cluster-heads. Then, a GW-D (Declare) message (Table V) is sent to its cluster-head and the neighbor cluster-head. By receiving this message, each of the cluster-heads updates its Neighbor CH Table (NCHT) (Table IV), in which it keeps the neighbor cluster-heads and its corresponding gateways. This process is illustrated in the MSC (Message Sequence Chart) diagram [26] in Fig. 6.

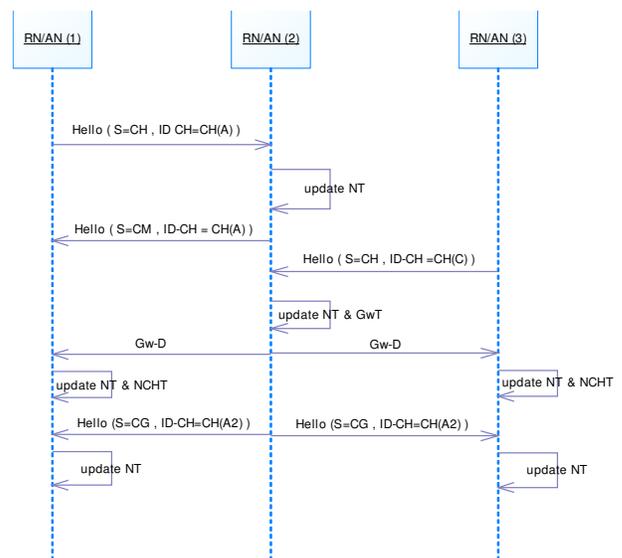


Figure 6. MSC of C-Gw selection scenario

A CM node may also become a D-Gw when receiving a Hello message from a CM belonging to another cluster, as illustrated in Fig. 7.

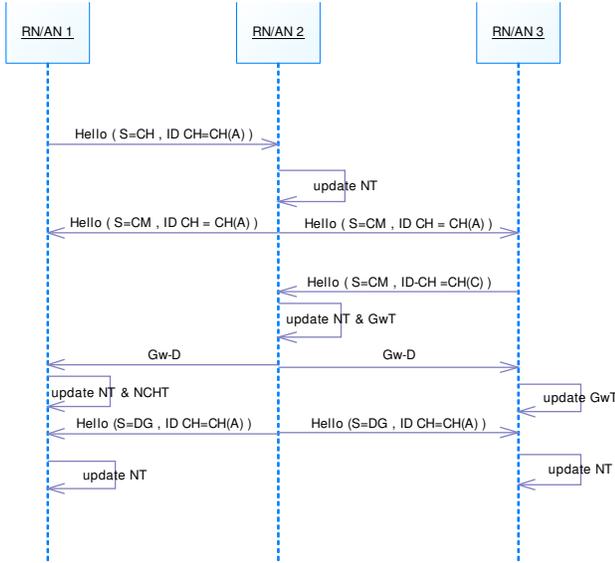


Figure 7. MSC of D-Gw selection scenario

TABLE III. GATEWAY TABLE (GWT)

Type-Gw	ID-CH1	ID-CH2	ID_D-Gw
---------	--------	--------	---------

TABLE IV. NEIGHBOR CLUSTER-HEAD TABLE (NCHT)

Neighbor ID-CH	Gw-ID	Type-Gw
----------------	-------	---------

TABLE V. GW-D MESSAGE

ID	Weight	Type-Gw	ID-CH	Neighbor ID-CH
----	--------	---------	-------	----------------

D. Intra-infrastruture Routing (IMRR)

Intra-Mesh Infrastructure Reactive Routing (IMRR) is the reactive routing sub-protocol of our proposed HQMR protocol. It is used to find routes in order to forward information between two nodes located within the infrastructure. It ensures QoS based routing for nodes belonging to a same cluster as well for those located in different clusters. Moreover, IMRR offers QoS guarantees by checking the QoS parameters namely bandwidth, delay and jitter at each node during the route discovery process.

Furthermore, the proposed IMRR sub-protocol is an enhancement of AODV routing protocol [27], which takes into account the clustering approach and the QoS verification in route discovery process.

1) IMRR operation

Fig. 8 illustrates the algorithm of IMRR operation. A node S starts directly to forward data if the destination D is one of its neighbors, with verified QoS or if a valid route to D exists in its routing table. Otherwise, S launches the route discovery process.

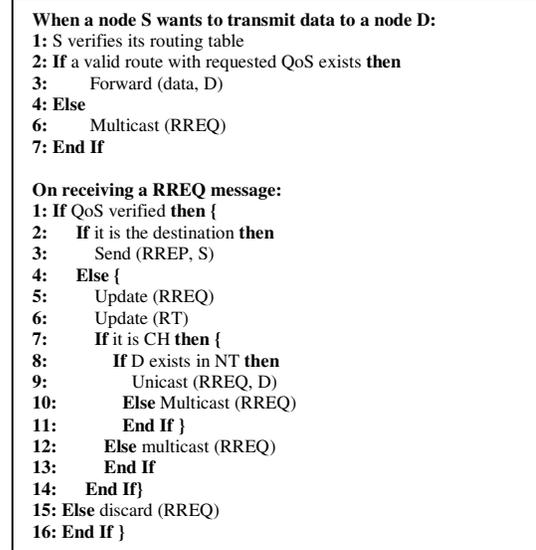


Figure 8. IMRR operation algorithm

The received RREQ message is either forwarded directly to the destination or forwarded to the multicast group formed by the different CHs, C-Gws, D-Gws and the MG. The use of the multicast group limits the broadcast of the RREQ messages, which helps reducing the load of the network.

According to this algorithm, two nodes from different clusters may communicate with each other only through a route formed by CHs and/or Gws and/or the Mesh Gateway.

An example of a communication between two nodes from different clusters is illustrated by a MSC diagram in Fig. 9.

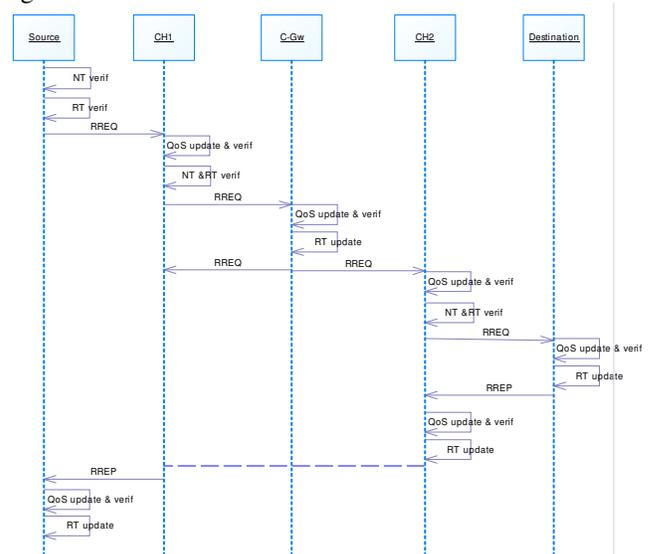


Figure 9. MSC for inter-clusters IMRR routing

2) Route Discovery Process

Like AODV protocol, IMRR uses RREQ message for route discovery (Table VI). However, the RREQ message used by our IMRR routing protocol introduces specific QoS fields to enable QoS based routing. Each intermediate node

proceeds to a QoS verification before forwarding the request (7).

($B_{off} \geq B_{req}$ or $B = \text{null}$) and ($D_{off} \geq D_{req}$ or $D = \text{null}$) and ($J_{off} \geq J_{req}$ or $J = \text{null}$) (7) where B is the bandwidth, D is the delay and J is the Jitter.

TABLE VI. RREQ MESSAGE

Src IP address	Dest IP address	Broadcast ID	Path	QoS Metric requested	QoS Metric offered	ID msg

In Fig. 10, we illustrate the processing of a RREQ message at each node. Unlike AODV protocol, only the destination node is able to respond to a RREQ message, so that it would have the entire path's estimated QoS to compare it properly to the requested one. Moreover, the duplicated RREQ messages (Broadcast ID already exists) are not rejected. Instead, we send as much as possible of RREQ messages to the destination to guarantee the discovery of the best path. In order to avoid an infinite loop of a message, each node verifies first if its address already exists in the Path field or not. Then, we introduce a new parameter called "ID msg" to distinguish the duplicate messages at a node. This parameter is updated at each intermediate node for each RREQ message received (duplicated or not). Thus, a node assigns a new "ID msg" for each request and inserts it into the RREQ message. Then, the reverse route is created within the routing table (Table VII), by taking into consideration this parameter, so that it would be used later for the RREP message forward.

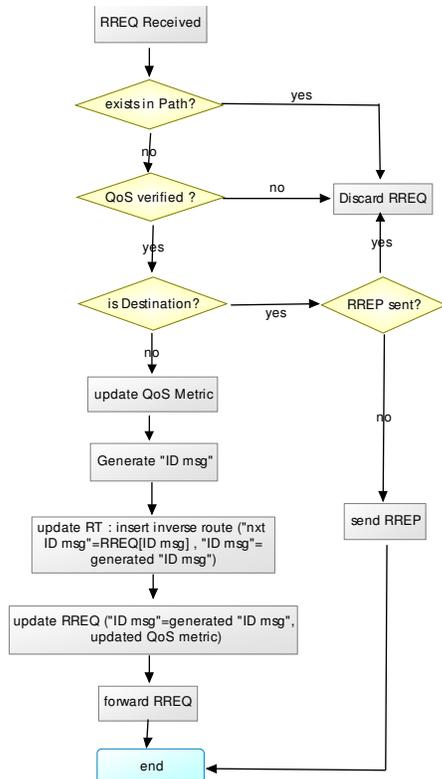


Figure 10. RREQ processing

TABLE VII. IMRR ROUTING TABLE

Dest IP address	Next Hop	Lifetime	QoS Metric offered	ID msg	Nxt ID msg

3) Route Reply Process

In order to establish a route toward the source node, the destination responds with a RREP message (Table VIII) to the first RREQ received verifying the requested QoS parameters and rejects the following RREQ messages. The processing of a RREP message at each intermediate node is illustrated by a flowchart in Fig. 11.

A mesh node determines the next hop thanks to the "ID msg" parameter. It updates then the routing table with the direct route and the "ID msg" with the "Nxt ID msg" of the routing table before forwarding the RREP message.

TABLE VIII. RREP MESSAGE

Src IP address	Dest IP address	Lifetime	QoS Metric requested	QoS Metric offered	ID msg

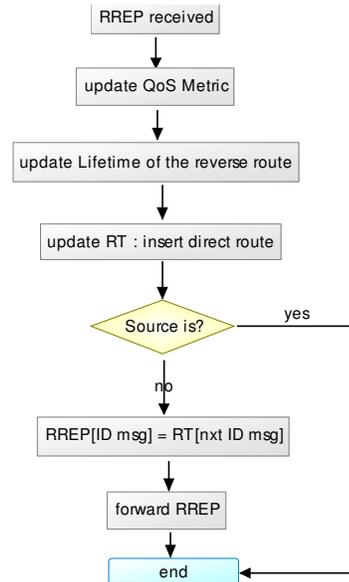


Figure 11. RREP processing

E. Inter-infrastructure Routing (IMPR)

Inter-infrastructure Mesh Proactive Routing (IMPR) is the second routing sub-protocol of HQMR, designed to ensure communications toward external networks, especially Internet network. Since most of the traffic goes through the Mesh Gateway to provide Internet services, we opted for a proactive tree based routing protocol, having the Mesh Gateway as a root and the different CHs and C-Gw and/or D-Gw as children. It is important to notice that the different cluster members would not participate in the trees construction process. In Fig. 12, we present the topology of the clustered infrastructure presented in Fig. 2, which we would have if we do not consider the different Cluster members and keep only the nodes that can play the role of children in our routing trees.

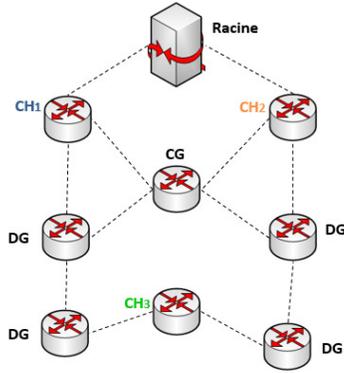


Figure 12. Network topology for trees construction

In addition, to provide QoS guarantees for real-time flows, IMPR deploys a multi-path routing concept to define three different routes, partially node-disjoint, between each child and the root. These routes would be used to construct three partially disjoint routing trees within the IEEE 802.16j wireless mesh infrastructure, in such a way that each tree is used to forward a specific type of traffic. To this end, we define for our protocol three service classes, namely interactive real-time applications class, Streaming applications class and Best Effort class. The first class is more sensitive to delay and jitter variations, the second one is more sensitive to jitter variation, and the last class is more exigent in terms of loss ratio. In other words, IMPR allows the construction of three partially disjoint trees with a common root: Real Time, Streaming, and Best Effort Trees.

1) Root Announcement process

The root (i.e., MG) broadcasts a RANN (Route Announcement) message to all its neighbors to announce its presence. This message is considered only by the CHs and the Gws. It is rejected by all the CM nodes. On receiving a RANN message (Table IX), each intermediate node stores the Path parameter in its route cache and updates it next by adding its address. It updates also the QoS Metric and proceeds to the forward of the updated RANN message to its multicast group formed by the CHs, the Gws, and the MG. In order to keep as many routes as possible, duplicated RANN messages are not rejected. Instead, to avoid an infinite loop of a message, each node verifies first if its address already exists in the Path field or not. In fact, each node keeps the entire path received through the RANN message in its route cache in order to be able to verify later the disjunction of two paths.

TABLE IX. RANN MESSAGE

Root IP address	Path	QoS Metric
-----------------	------	------------

2) Routing trees construction

Each node waits for a certain time T_s before starting the routing trees construction process, in order to store the maximum of paths. Firstly, using the routes selection algorithm (Fig. 13), each node selects a route for the Real Time Tree. This route is validated as one of the tree branches by an exchange of PREQ and PREP messages with the root.

Once the PREP received from the root, each node removes the chosen path from its route cache and starts the construction process of the second routing tree in the same manner. Then, the mechanism is repeated for the third routing tree. In fact, the exchange of PREQ/PREP messages performed for routes validation is used to ensure that each intermediate node of a path is using the same path toward the root, so that each node has no more than a single branch toward the root of a tree.

3) Routes Selection Algorithm

This algorithm is described in Fig. 13. The idea is to select at each node a potential path for each routing tree, satisfying the requirements of the defined service classes. For the first path corresponding to the Real Time Tree, we choose the best in terms of delay and jitter with satisfying bandwidth metric. The second one should be partially disjoint from the first one to reduce congestion issues, with good values of the jitter QoS parameter. Lastly, from the remaining paths, we select the best in terms of disjunction over the other paths.

Some nodes may not be able to select three different paths. Thus, for the case where a node has only selected two paths, the first one would be used to forward the highest priority traffic, while the second one would be shared between the two other service classes. If only one path is present at a node, we adopt the default QoS mechanism of IEEE 802.16j to share it between the three service classes.

```

P ← set of stored paths ; Disj ← number of common nodes between paths
HC: Hop Count ; wi : QoS parameters' weight ; L: weight of a path

1: If treei = 1 then
2:   A = {P}(D<Dmax and J<Jmax)
3:   If A ≠ ∅ then
4:     P1 = minHC { maxBw A }
5:   Else
6:     B = {P}(D<Dmax)
7:     If B ≠ ∅ then
8:       Calculate L = w1*rankdescBw + w2*rankascJ for each path in B
9:       P1 = minL B
10:    Else
11:      Calculate L = w1*rankdescBw + w2*rankascD + w3*rankascJ for each Path in P
12:      P1 = minL P
13:    End If
14:  End If
15: End If
16: If treei = 2 then
17:   P = P \ {P1}
18:   A = {P}(J<Jmax)
19:   If A ≠ ∅ then
20:     Calculate L = a*rankdesc Bw + b*rankasc Disj + c*rankasc J for each Path in A
21:     P2 = minL A
22:   Else {
23:     L = w1*rankdesc Bw + w2*rankasc J + w3*rankasc Disj for each Path in P
24:     P2 = minL P
25:   End If
26: End If
27: If treei = 3 then
28:   P = P \ {P1} ; P3 = minHC { minDisj P }
29: End If

```

Figure 13. IMPR routes selection algorithm

4) Path Request Process

By executing the route selection algorithm, a node selects a path for its i^{th} routing tree and sends a PREQ message (Table X). Each intermediate node compares its chosen path

for its i^{th} routing tree to the path carried by the PREQ message. If the next hop in the two paths is different, the node either modifies its entire path or updates the path in the PREQ message, as presented in the Flowchart in Fig. 14. Then, the intermediate node updates its routing table (Table XI) with both the direct route (toward the root) and the reverse route (toward the source) and forwards the PREQ message to the next hop.

TABLE X. PREQ MESSAGE

Src IP address	Dest IP address	Path	ID-Path	Level*
----------------	-----------------	------	---------	--------

*- Level : the level of a node in the Real Time tree

TABLE XI. IMPR ROUTING TABLE

Dest IP address	Next Hop	ID-Path
-----------------	----------	---------

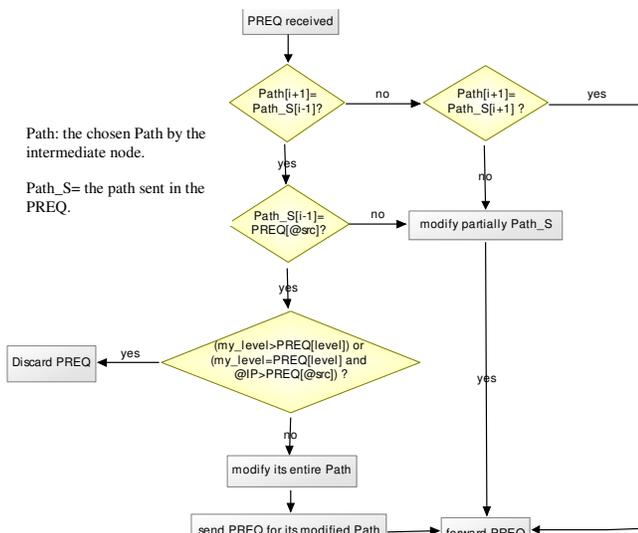


Figure 14. Flowchart of PREQ process

5) Path Replay Process

On receiving the PREQ message, the root updates its routing table and sends a PREP (Table XII) message to its child.

TABLE XII. PREP MESSAGE

Dest IP address	Path	ID-Path
-----------------	------	---------

Each intermediate node adds its address to the Path parameter of the PREP message and forwards it to the destination. Once the destination receives the PREP message, it updates its routing table and its chosen path for the routing tree if it is different from the Path parameter in the PREP message. Then, it removes it from its route cache to begin the selection of a route for the next tree.

VI. HQMR USAGE SCENARIOS

In this section, we present two different usage scenarios of our HQMR protocol, describing how a path is selected to reach a destination within or outside the mesh infrastructure.

A. Intra-infrastructure Routing Usage Scenario

This scenario describes how to determine a QoS verified path between two nodes from different clusters for a VoIP application between two mesh clients of our architecture. To this end, the reactive routing bloc, named IMRR would be used and a RREQ message is generated for route discovery process. In Fig. 15, we illustrate the RREQ process through each intermediate node by comparing the offered QoS to the requested one ($B_{\text{req}}=56\text{Kb/s}$, $D_{\text{req}}=150\text{ms}$ and $J_{\text{req}}=20\text{ms}$).

The first RREQ received by D ($\langle 2, 155, 19 \rangle$) does not satisfy the requested delay parameter. Thus, this message is discarded and D waits for another RREQ messages. Since the second message received (RREQ2) verifies the different QoS parameters ($\langle 2, 145, 13 \rangle$), a RREP message is unicasted to the source node.

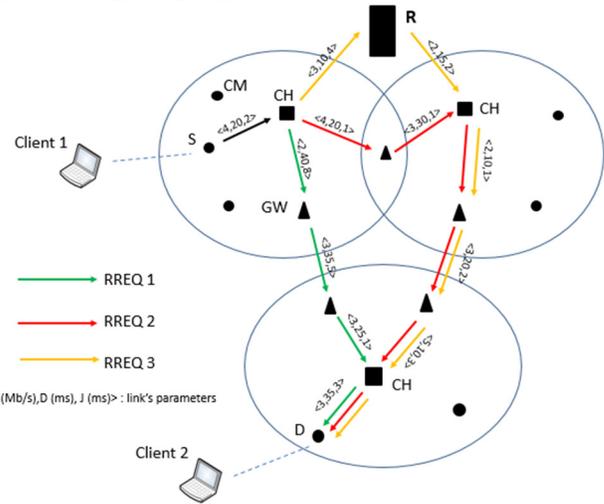


Figure 15. Intra-infrastructure usage scenario

Then, regarding the third RREQ message received, it would be discarded since a RREP message has been already sent back. By this way, the route discovered by RREQ2 would be used to forward the traffic of the VoIP application between the two mesh clients.

B. Inter-infrastructure Routing Usage scenario

For communications with the Internet network, the proactive routing protocol IMPR of HQMR protocol is used. In this scenario, we describe how to forward a VoD (Video on Demand) application traffic from a streaming video server in the Internet. To this end, three QoS based routing trees are constructed. Fig. 16 shows the topology used for this scenario. Then, we illustrate in Fig. 17 the three routing trees built over this topology to forward traffic to the Internet networks.

For the first routing tree, by executing the route selection algorithm ($D_{\text{max}}=150\text{ms}$, $J_{\text{max}}=20\text{ms}$), we chose the paths with satisfying delay and Jitter parameters. To better explain the construction process of this tree at each node, we describe for example the case of the node B. We have four paths towards the root satisfying the delay and jitter parameters: B-A-R: $\langle 4,30,11 \rangle$; B-C-R: $\langle 3,70,6 \rangle$; B-D-A-R: $\langle 3,95,16 \rangle$; B-E-C-R: $\langle 2,70,9 \rangle$. Then, the path with the

highest Bandwidth is selected: B-A-R. This process is repeated at each node of the topology and a PREQ message is sent for route validation. Once a node receives a PREP message for the first route, it starts the selection of a path for the second routing tree.

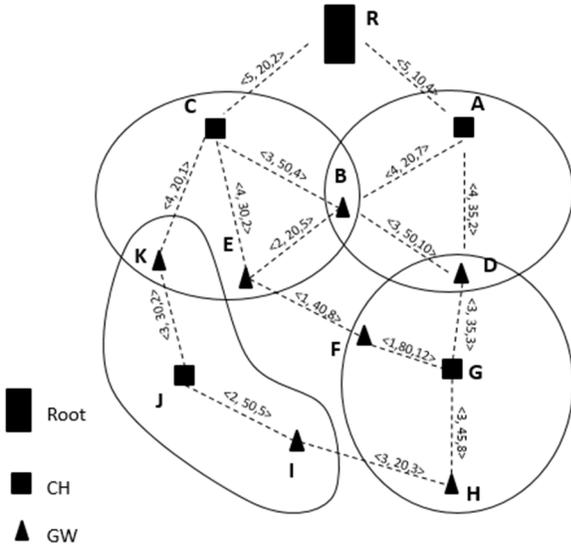


Figure 16. Example of clustered wireless mesh infrastructure

Similarly, the path at each node for the second routing tree is selected according to the routes selection algorithm. For example, at the node E, we have two paths verifying the jitter parameter: E-B-C-R: $\langle 2, 90, 11 \rangle$; E-B-A-R: $\langle 2, 50, 16 \rangle$. The second path is the one selected by E since it offers better disjunction with the path selected for the first tree. However, the PREQ of this message would be changed at the node B. In fact, the nodes B and E have the same level parameter but node B has a greater IP address than E. Thus, the path in the PREQ sent by the node E would be changed (to E-B-C-R) to correspond to the route chosen by the node B: B-C-R.

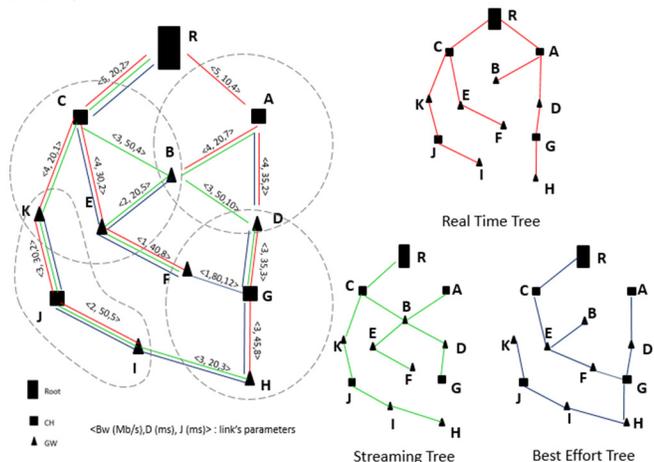


Figure 17. The QoS based routing Trees

By receiving the PREP message for the selected route, each node starts the selection of its third route that is the most disjoint route to the two first selected paths with a minimum of hops. For example, the node I according to these conditions chooses the path I-H-G-D-A-R. However, its PREQ at the node G would be changed (see the flowchart in Fig. 14). The selected path by the node I would be modified partially (I-H-G-F-E-C-R) to correspond to the one selected by the node G. Then, the set of paths selected at each node forms the routing tree for the third service class.

Regarding our usage scenario, and in order to forward the VoD traffic, the second routing tree would be used since this application is considered as an application of the Streaming service class.

VII. PERFORMANCE EVALUATION OF IMRR

To evaluate the performance of our routing protocol, we have developed the source code of HQMR protocol using the network simulator ns-3. Then, we have conducted some simulation scenarios to compare our IMRR routing sub-protocol and the AODV protocol in a wireless mesh environment.

In this section, we define briefly the used network simulator namely ns-3. Then, we present the different performance parameters that we have evaluated as well as the simulation environment.

A. Network Simulator Ns-3

Ns-3 [28] is a discrete-event network simulator, developed within the ns-3 project, started in 2006 to eventually replace the aging ns-2 simulator. It is a free open software, which can be used for research to contribute and share the developed codes. The ns-3 software is built on C++, containing a set of network simulation modules implemented as C++ objects. Then, the simulation scripts may be written in C++ or Python.

The existing library of modules allows simulating popular wireless networks (e.g., Wi-Fi, WiMAX, LTE) in a simple way. Moreover, thanks to the availability of the source code of existing modules, it is possible to modify the operation of any module from the library. It is also possible to create new modules, implementing algorithms or protocols not present in the existing library.

B. Performance metrics

In our simulations, we have measured the following key performance parameters to evaluate the effectiveness of our IMRR routing sub-protocol.

- Routing Overhead: the total number of control messages generated during the simulation for route discovery process. For messages sent over multiple hops, each transmission (each hop) counts as one generated control message.
- Route Discovery Convergence: the necessary time to establish the route between the source node and the destination one. It is the time between the received RREP message and the RREQ message being sent by the source.

- Average Throughput: It is the mean of the number of packets successfully transmitted to their final destination per unit time.
- Average End-to-End Delay: This is the overall average delay required by a packet to travel from source node to its destination node. It includes all possible delays caused by queuing delay, retransmission delays and propagation delay.
- Average Jitter: It is the mean of the difference between the end-to-end delay values.

C. Simulation Environment

The simulation environment consists up to 25 stationary mesh nodes arranged in a grid topology. Simulation time is 50 seconds. Each scenario is simulated five times and an average value is taken for the performance analysis. Table XIII shows the used simulation parameters.

TABLE XIII. SIMULATION PARAMETERS

Simulation parameters	Value
Routing Protocols	IMRR & AODV
Simulation Time	50s
Nodes Number	6 to 25 nodes
Mobility Model	GridPositionAllocator / static
Traffic Model	CBR (UDP) / VoIP (UDP)
Packet Size	512 bytes / 160 bytes
DataRate	512kbps / 64 kbps

To analyze the different performance metrics, two different traffic models are used in the elaborated scenarios. The first one is a generic CBR traffic used in scenario 1 (see Section VIII.A) and the second one simulates a VoIP traffic used in scenario 2 (see Section VIII.B).

VIII. SIMULATION RESULTS AND ANALYSIS

A. Scenario 1

To evaluate our protocol performance in terms of routing overhead and route discovery convergence, we perform different simulations, by varying the number of nodes within our mesh infrastructure, while considering a constant-bitrate (CBR) traffic. This traffic is modeled with 512-byte data packets and a data rate of 512kbps.

In fact, the routing overhead for route discovery process includes, for both AODV protocol and IMRR sub-protocol, the Hello messages, the RREQ messages and the RREP messages. Figure 18 shows the variation of the number of RREQ messages according to the number of nodes in the network. We notice that the amount of RREQ messages of HQMR protocol increases with the number of nodes while AODV protocol presents a low variation. This is explained by the fact that AODV protocol rejects each duplicated RREQ message at each node. By this way, less RREQ messages are forwarded in the network. However, in order to ensure the best route discovery in terms of QoS guarantee, HQMR does forward some duplicated received RREQ

messages to the destination (up to three at each node). On the other hand, AODV presents a higher amount of RREP and Hello messages, as illustrated in Fig. 19 and Fig. 20. Indeed, we observe a major difference between the two protocols in terms of RREP messages since AODV protocol allows that an intermediate node replies to a RREQ message. However, with the HQMR protocol, only the destination is allowed to reply to a RREQ message after verifying the offered QoS parameters of the entire received route. Moreover, each time a node sends a RREQ message, sending a Hello message is deferred. Thus, the amount of forwarded Hello messages is inversely proportional to the amount of RREQ messages, which explains the results illustrated in Fig. 18 and Fig. 20.

Since the HQMR protocol forwards less RREP messages than the AODV protocol for all network sizes and since the amount of Hello messages is offset by the RREQ messages and inversely, we can conclude, from these results, that the IMRR sub-protocol has a better global routing overhead.

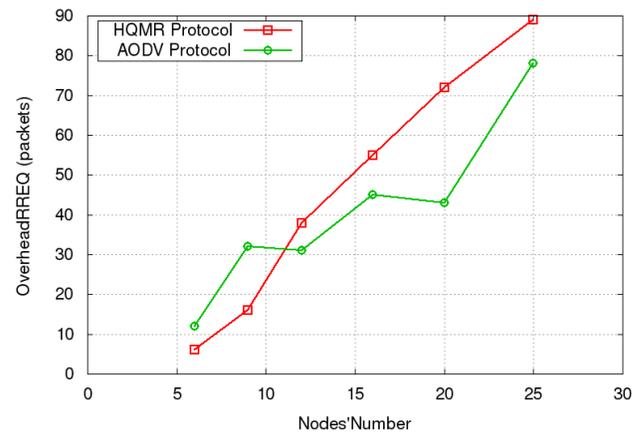


Figure 18. RREQ Messages Overhead

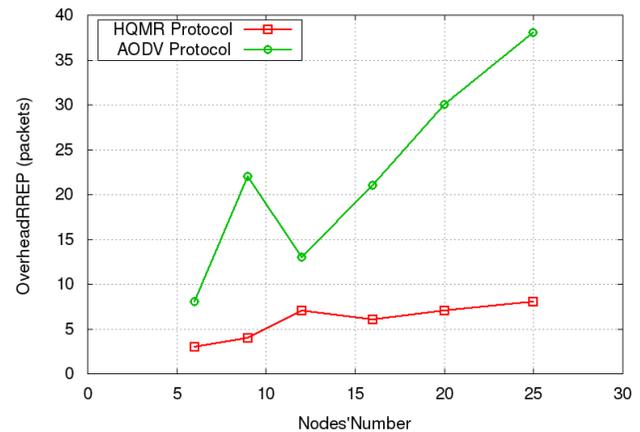


Figure 19. RREP Messages Overhead

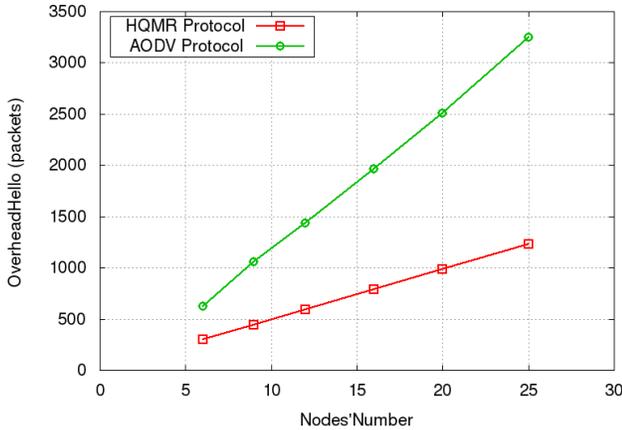


Figure 20. Hello Messages Overhead

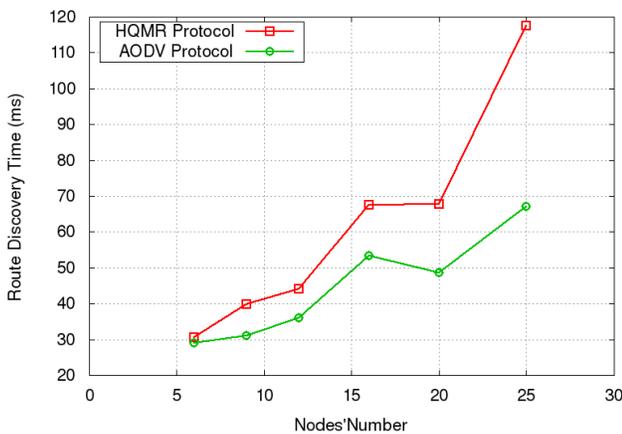


Figure 21. Route Discovery Convergence

Figure 21 illustrates the variation of the route discovery time performance parameter according to the size of the mesh topology. For both protocols, the route discovery convergence time increases with the number of nodes since the discovered route may be longer in terms of hops. Although IMRR sub-protocol has higher values than AODV, we perceive a close variation with the AODV values. Actually, AODV uses the minimum hops number as a metric. That is why, AODV spends less time in discovering routes. However, HQMR protocol accepts longer routes as it offers better QoS parameters to ensure a better routing for real-time and streaming applications.

B. Scenario 2

To underline the effectiveness of our routing protocol in route discovery enabling to correctly forward real-time applications, we evaluate the corresponding performances of such applications in terms of average throughput, average end-to-end delay and average jitter parameters within a mesh topology, since this type of application, i.e., interactive applications, is very sensitive to delay and jitter variation.

To simulate a voice conversation, we used a traffic pattern corresponding to the G711 encoder, which produces 50 packets per second with 160 bytes of payload each. Then,

we have introduced a noise over some links to simulate network perturbation. The simulations are conducted to compare the IMRR sub-protocol and the AODV protocol by varying the network size.

The comparative results of throughput, end-to-end delay and jitter QoS parameters are shown in Figs. 22, 23, and 24, respectively. We observe that HQMR offers better values of the average throughput than the AODV protocol for the different mesh network sizes. Besides, we notice a considerable difference concerning the variation of the delay and jitter parameters. The corresponding values while using AODV are more than twice the QoS values while using HQMR.

Thus, the HQMR protocol offers a better route than AODV, especially in terms of delay and jitter, to forward VoIP traffic. Actually, the AODV protocol does not take into consideration the state and the QoS offered by the different links within the mesh topology. It is based only on the number of hops for its route selection. On the other hand, HQMR process a QoS verification during the route discovery process to determine the route satisfying the requested QoS parameters depending on the type of the application to forward.

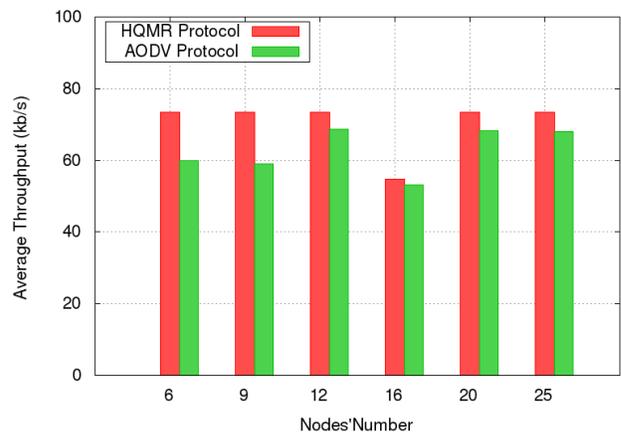


Figure 22. Average Throughput of a VoIP Application

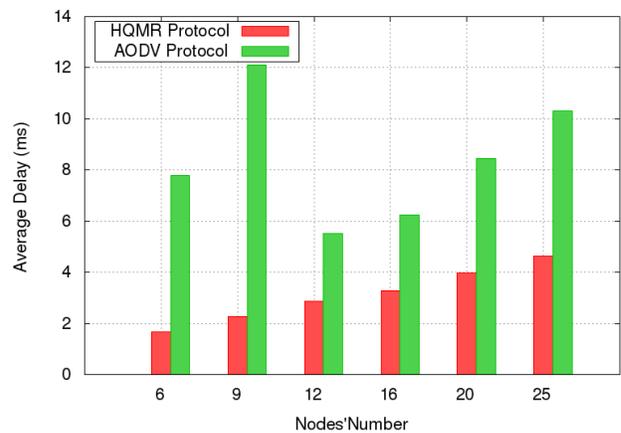


Figure 23. Delay Evaluation of a VoIP Application

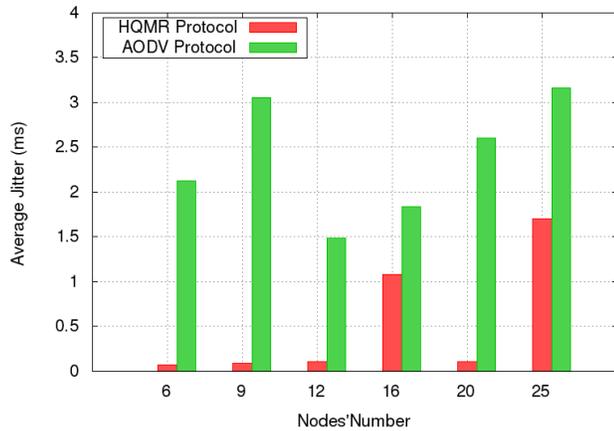


Figure 24. Jitter Evaluation of a VoIP Application

IX. CONCLUSION

In this paper, we presented our proposed hybrid wireless mesh architecture composed of two different domains: an IEEE 802.16j-based infrastructure domain and several IEEE 802.11s based client domains. Then, we have specified the HQMR protocol for ensuring routing functionalities within the wireless mesh infrastructure of our global architecture. It is a hybrid QoS based routing protocol formed by a reactive routing sub-protocol for a clustered infrastructure and a proactive multipath tree based routing sub-protocol for communications toward Internet network. Two usage scenarios are presented to show the importance of HQMR in order to provide real time and streaming applications with QoS guarantee in wireless mesh networks. Then, we presented the different simulations scenarios conducted to evaluate the performance of our IMRR routing sub-protocol in terms of routing overhead and route discovery convergence time, as well as the performance of a real-time interactive application (VoIP) in terms of average throughput, average end-to-end delay and average jitter while using our routing protocol in a mesh topology.

ACKNOWLEDGMENT

This work has been funded by the Regional Council of Burgundy, France.

REFERENCES

- [1] H. Bargaoui, N. Mbarek, O. Togni, and M. Frikha, "Hybrid QoS Based Routing for IEEE 802.16j Mesh Infrastructure," AICT 2014, The Tenth Advanced International Conference on Telecommunications, 2014, pp. 110-118.
- [2] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Comput. Netw.*, vol. 47, n° 4, pp. 445-487, March 2005.
- [3] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Commun. Mag.*, vol. 43, n° 9, pp. S23-S30, 2005.
- [4] "IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking," *IEEE Std 80211s-2011 Amend. IEEE Std 80211-2007 Amend. IEEE 80211k-2008 IEEE 80211r-2008 IEEE 80211y-2008 IEEE 80211w-2009 IEEE 80211n-2009 IEEE*

- 80211p-2010 IEEE 80211z-2010 IEEE 80211v-2011 IEEE 80211u-2011*, pp. 1-372, 2011.
- [5] "IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems Amendment 1: Multihop Relay Specification," *IEEE Std 80216j-2009 Amend. IEEE Std 80216-2009*, pp. 1-290, 2009.
- [6] "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1," *IEEE Std 80216e-2005 IEEE Std 80216-2004Cor 1-2005 Amend. Corrigendum IEEE Std 80216-2004*, pp. 0_1-822, 2006.
- [7] Q. Xue and A. Ganz, "QoS Routing for Mesh-Based Wireless LANs," *Int. J. Wirel. Inf. Netw.*, vol. 9, n° 3, pp. 179-190, July 2002.
- [8] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," New York, NY, USA, 2004, pp. 133-144.
- [9] V. Kone, S. Das, B. Y. Zhao, and H. Zheng, "QUORUM: quality of service in wireless mesh networks," *Mob Netw Appl*, vol. 12, n° 5, pp. 358-369, December 2007.
- [10] L. Liu, L. Zhu, L. Lin, and Q. Wu, "Improvement of AODV Routing Protocol with QoS Support in Wireless Mesh Networks," *Phys. Procedia*, vol. 25, pp. 1133-1140, 2012.
- [11] Y. Zhang, Y. Wei, M. Song, and J. Song, "R-AODV: Rate aware routing protocol for WiFi mesh networks," 2006 IET International Conference on Wireless, Mobile and Multimedia Networks, 2006, pp. 1-4.
- [12] X. Cheng, P. Mohapatra, S. Lee, and S. Banerjee, "MARIA: Interference-Aware Admission Control and QoS Routing in Wireless Mesh Networks".
- [13] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 15, n° 7, pp. 1265-1275, 1997.
- [14] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan, "A cluster-based approach for routing in dynamic networks," *SIGCOMM Comput Commun Rev*, vol. 27, n° 2, pp. 49-64, April 1997.
- [15] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Clust. Comput.*, vol. 5, no 2, pp. 193-204, April 2002.
- [16] Z. Haas, M. Pearlman, and P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks". <http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-04>. Consulted : 2015-05-18.
- [17] M. Singh, S. G. Lee, T. W. Kit, and L. J. Huy, "Cluster-based routing scheme for Wireless Mesh Networks," 13th International Conference on Advanced Communication Technology (ICACT), 2011, pp. 335-338.
- [18] D. Kaushal, A. G. Niteshkumar, K. B. Prasann, and V. Agarwal, "Hierarchical Cluster Based Routing for Wireless Mesh Networks Using Group Head," International Conference on Computing Sciences (ICCS), 2012, pp. 163-167.
- [19] Jack Tsai and Tim Moors, "A Review of Multipath Routing Protocols: From Wireless Ad Hoc to Mesh Networks," ACoRN Early Career Res. Workshop Wirel. Multihop Netw., 2006.
- [20] R. Krishnan and J. A. Silvester, "Choice of allocation granularity in multipath source routing schemes," in *Networking: Foundation for the Future*, IEEE INFOCOM '93. Proceedings, Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies, 1993, pp. 322-329, vol.1.
- [21] N. S. Nandiraju, D. S. Nandiraju, and D. P. Agrawal, "Multipath Routing in Wireless Mesh Networks," IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), 2006, pp. 741-746.
- [22] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MobiHOC, 2000, pp. 3-10.

- [23] X. Hu and M. J. Lee, "An efficient multipath structure for concurrent data transport in wireless mesh networks," *Comput. Commun.*, vol. 30, no 17, pp. 3358-3367, November 2007.
- [24] M. S. Siddiqui, S. O. Amin, J. H. Kim, and C. Hong, "MHRP: A Secure Multi-Path Hybrid Routing Protocol for Wireless Mesh Network," *IEEE Military Communications Conference, MILCOM, 2007*, pp. 1-7.
- [25] "IEEE Draft Standard for Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-Amendment 8: IEEE 802.11 Wireless Network Management," *IEEE P80211vD160*, pp. 1-428, November 2010.
- [26] D. Harel and P. S. Thiagarajan, "Message Sequence Charts," in *UML for Real*, L. Lavagno, G. Martin, et B. Selic, Éd. Springer US, 2003, pp. 77-105.
- [27] "Ad hoc On-Demand Distance Vector Routing (AODV)". <http://www.ietf.org/rfc/rfc3561.txt>. Consulted : 2015-05-18.
- [28] "ns-3". <http://www.nsnam.org/>. Consulted : 2015-05-18.

On the Performance of a Flow Aggregation Scheme for Seamless QoS and Utility Oriented Mobility Support in Wireless Mesh Networks

Salvatore Vanini*, Dario Gallucci*, Silvia Giordano*, Maciej Urbanski†, and Przemyslaw Walkowiak†

*Information Systems and Networking Institute
SUPSI, Manno, Switzerland

Email: [salvatore.vanini,dario.gallucci,silvia.giordano]@supsi.ch

†Institute of Control and Information Engineering
Poznan University of Technology, Poznan, Poland

Email: [maciej.urbanski,przemyslaw.walkowiak]@put.poznan.pl

Abstract—Wireless Mesh Networks are a viable solution for extending the coverage of networks without the need to expand their static infrastructure. This is achieved by employing wireless nodes that can share their network resources. Unfortunately, building wireless mesh networks is still far away from reality due to the lack of mechanisms to meet the Quality of Service (QoS) expectations of mobile users running heterogeneous applications. To address this issue, in this paper we describe a modification of the WiOptiMo framework, which is a solution originally designed for seamless handover management in the Internet. Specifically, QoS support is provided by aggregating application traffic flows with the same characteristics to limit overhead and by relaying compressed aggregated flows to the appropriate mobility provider. Evaluation on a real wireless mesh network testbed showed that this scheme is noteworthy in terms of link utilization, improved QoS and performance against Mobile IPv6, which is the standard for enabling mobility in IP networks. To show its adaptability, we present a scenario where WiOptiMo can be employed to support network mobility when the range of a network is extended by exploiting neighboring nodes. Beside the evidence of performance gain against Mobile IPv6, the other contribution of this paper is the proposal of an incentive mechanism to motivate users of a wireless mesh network to share their network resources with other nodes, which rewards them based on the amount of bytes saved thanks to the use of the WiOptiMo flow aggregation scheme.

Keywords—Wireless Mesh Networks; Seamless Handover; QoS Mobility Support; Flow Aggregation; Flow Classification; Mobile IP; Network Utility Maximisation.

I. INTRODUCTION

This paper builds on the work [1] (presented at MOBILITY 2014), which dealt with the design, implementation and evaluation of a flow classification and aggregation scheme for managing multiple applications with different Quality of Service (QoS) requirements in Wireless Mesh Networks (WMNs).

Recent years have witnessed a significant reduction in the costs of mobile computing platforms (e.g., laptops and smartphones), especially the hardware used in WiFi devices and has led to a widespread use of WMNs. WMNs provide multiple services to people using their mobile devices via a combination of fixed and mobile nodes, interconnected via wireless links to form a multi-hop ad-hoc network. WMNs are a cost-effective solution to extend the range of wired infrastructure networks with the help of easy to deploy wireless nodes. For example, the backbone of a telecom service provider can be easily expanded utilizing mechanisms to manage resources

of wireless nodes [2][3]. Existing mechanisms work only in scenarios where wireless connection stability can be ensured. For example, CARMNET [4][5] utilizes the WMN paradigm to enable nearby wireless devices communicate with each other and proposes a distributed resource management method that can be easily integrated with a telecom IP-based Multimedia Subsystem (IMS) software infrastructure. This method (implicitly) assumes that the underlying network connectivity is not affected by topological changes (e.g., gateway changes) caused by the mobility of network's nodes. During those changes, packets for a given application flow might be rejected because of the change of the IP address, or they might be lost due to out-dated routing information. As a consequence, the quality and performance of correspondent applications can significantly decrease. The main protocols that can be used for supporting network mobility in IP-based WMN architectures are Mobile IPv4 (MIPv4) [6] and Mobile IPv6 (MIPv6) [7]. Mobile IP focuses on keeping the IP identity of a mobile node. MIPv6 is an enhancement of MIPv4 in terms of performance, however it might have long delay (handover latency) and high packet loss rate because of signaling traffic overhead. For this reason, several extensions to the MIPv6 base protocol, such as Fast Handovers for MIPv6 [8] and Hierarchical MIPv6 [9], have been proposed to get a better performance. A completely different approach to network mobility support is the use of schemes adopted in pure ad-hoc networks, which focus on rerouting (i.e., finding an alternative path in a timely manner, so that a flow can be handed off to the new path upon link disruption). Unfortunately, these mechanisms performs poorly in WMNs. To overcome the limitations of these approaches, several works have proposed different schemes for providing QoS and seamless mobility support in WMNs. However, many of them are not designed to manage multimedia services with QoS requirements—e.g., Voice over IP (VoIP) or Video on Demand (VoD).

This work is organized as follows. In Section II, we describe the previous work on mobility management in WMNs. In Section III, we present an extension of our WiOptiMo [10] framework to provide generalized QoS mobility support in WMNs. In Sections IV and V, we describe our enhanced framework and flow aggregation scheme to provide the required QoS to different types of applications in a WMN scenario. Then, in Section VI, we evaluate the performance improvement with respect to its typical configuration for WMNs. In Section VII, we compare the performance of

WiOptiMo with the standard MIPv6 implementation. Then, in Section VIII, we briefly describe the CARMNET WMN architecture - which provides mechanisms for sharing network access and extending network coverage - and present a scenario where WiOptiMo is used to bridge the signal gap between different access points and to provide handover functionality. Finally, in the context of a CARMNET WMN architecture, we propose an incentive mechanism for users to share their network access, which is based on the amount of bytes saved thanks to the use of the WiOptiMo aggregation scheme.

II. RELATED WORK

The existing work on mobility management in WMNs focuses on providing network-layer mobility support. RFC 4886 [11] specifically addresses the issue of network mobility. The main protocol used for mobility management at the IP layer is MIPv6 [7]. However, MIPv6 has some well known drawbacks such as signaling traffic overhead. This results in long delay (handover latency) and high packet loss rate, thereby causing a QoS deterioration of real-time traffic. Furthermore, MIPv6 has some scalability problems that arise since it handles mobile node local mobility in the same way as global mobility. For these reasons, several extensions of MIPv6, such as Fast Handovers for MIPv6 [8] and Hierarchical MIPv6 [9], have been introduced to increase its performance. Fast Handovers for MIPv6 was proposed to reduce the handover latency by providing IP connectivity as soon as a mobile node attaches to a new subnet. To realize this, a mobile node performs a probe task to discover nearby access points. The main drawback of this process is that the mobile node cannot receive or send data during the probe phase. HMIPv6 was proposed to handle handover locally, thereby reducing unnecessary signaling overhead and latency within a domain, but suffering from same delay for global communication. To sum up, despite MIPv6 extensions, mobility management with QoS provision in WMNs remains a challenging task.

Interworking between 3GPP cellular network and WLAN is addressed by the Third Generation Partnership Program (3GPP), which developed an architecture to enable 3GPP cellular network subscribers to access WLAN service [12]. The interworking architecture provides fast deployment for global roaming and billing. This initiative is focused on specific standards and its standardization is currently ongoing [13].

The different solutions presented in literature focus on managing the address of a mobile node due to the handover process. In general, we can distinguish between intra-domain and inter-domain mobility. The first refers to handovers inside the same network domain, the second to handovers between different network domains. MobileNAT [14] addresses both intra- and inter-domain mobility. It allows a mobile node to keep a fixed IP address as it roams across the same or a different domain. MobileNAT requires a modification at the network layer stack of a mobile node and changes to the standard DHCP protocol, which introduces network latency. SyncScan [15] is a Layer-2 procedure for intra-domain handover in 802.11 infrastructure mode networks. It achieves good performance at the expense of a required global synchronization of beacon timings between clients and access points (AP). iMesh [16] provides low handover latency for Layer-3 intra-domain handovers between APs of a WMN. However, the handover latency depends on the number of

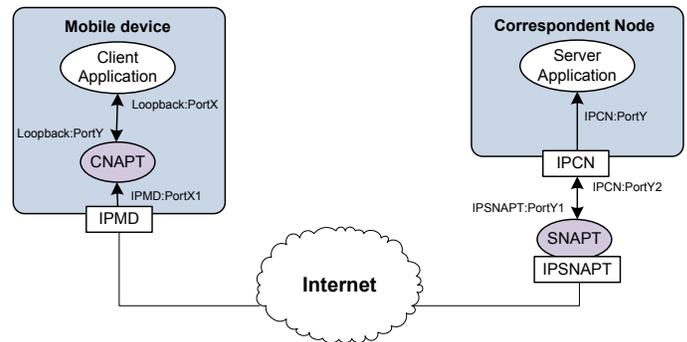


Figure 1. WiOptiMo's CNAPT and SNAPT IP decoupling.

nodes between the new and old AP. BASH [17] focuses on the design of an intra-domain Layer-2 seamless handover scheme for 802.11 WMNs, but the handover protocol requires modifications at every mobile client. Authors of [18] use tunneling, as well as the standard Mobile IPv6 solution [7] and most of the existing network-layer mobility management schemes based on Mobile IP, such as Mobile Party [19] and AODV-PRD [20]. Tunneling introduces extra delay for the encapsulation/decapsulation of packets and has intrinsically low flexibility. Finally, SMesh [21] provides a 802.11 mesh network architecture for both intra-domain and inter-domain handovers. For intra-domain handovers, SMesh generates high network overhead, which grows linearly with the number of mobile clients. In case of inter-domain handovers, network overhead generated by SMesh is proportional to the number of connections of a mobile client. The WiOptiMo framework provides mobility support by separately managing each application's flow, to meet the QoS expectations of all applications. In [10], we describe the architecture of WiOptiMo and in [22] we present how it is adapted to handle a WMN context. In the next sections, we show how its architecture has been modified to handle efficiently multiple application's flows with different QoS requirements and improve performance of standard mobility management mechanisms.

III. THE WIOPTIMO FRAMEWORK

WiOptiMo handles IP network mobility and enables handovers initiated by a mobile device. It manages the mobility of every device with the help of two software modules: Client Network Address & Port Translator (CNAPT) and Server Network Address & Port Translator (SNAPT). Together, these two components provide decoupling between the IP address assigned to a mobile device and the IP address used to access a service on the Internet. CNAPT and SNAPT hide any change of the IP address when a mobile host moves between different access networks, inside the same domain or between different domains. In Figure 1, we describe the basic idea of the WiOptiMo framework. A mobile device with IP address IPMD has an active TCP session to a corresponding node with IP address IPCN. The TCP data packets are first relayed to the local CNAPT, which in turn relays them to the SNAPT. Upon receiving packets, the SNAPT (processes and) forwards them to the IPCN address. When the mobile device moves to a new network and gets a new IP address, the change in IP address does not affect the application layer because the application packets are sent to the the local CNAPT, which relays them to

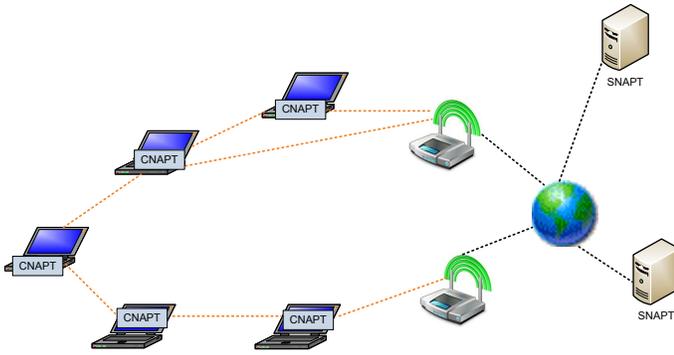


Figure 2. WiOptiMo configuration for a WMN.

the SNAPT with fixed IP address (IPSNAPT). This mechanism also allows a mobile node of a WMN to change gateway transparently (e.g., when node moves out of the reach of the initial gateway due to the mobility of the associated user), without suffering service disruption. To correctly manage the handover process, CNAPT and SNAPT exchange handshaking packets with each other using a control socket.

In a generalized setting, mobile devices have CNAPT installed on them, while an Internet server or any node in a network (as in the scenario previously described) have SNAPT installed on them.

A. WiOptiMo Architecture for a WMN

In [23], we present a general configuration of our WiOptiMo for a WMN. We exploit the flexibility of location where a SNAPT can be installed to address scalability issues that might arise in a WMN. In this scenario, multiple SNAPTs can be deployed on mesh routers or on Internet nodes to avoid network congestion in a single spot. This solution overcomes the scalability issues of MIPv6 because local mobility can be managed in a more efficient way. Every mobile wireless device has CNAPT installed on it to provide independent mobility support. We use a combination of network status monitoring and user configurable policy to enable every CNAPT to choose a suitable SNAPT that will relay its application flows. At startup, each CNAPT connects to a fixed SNAPT specified in a configuration file. Then, it receives a list of other available SNAPTs from the currently connected SNAPT, and measures the delay towards them by means of passive and active monitoring of the control connection towards the SNAPTs, used for handshaking. CNAPTs also take into account the bandwidth used by applications in order to make a more wise SNAPT choice. The CNAPTs select a SNAPT to relay their data depending on the measured delay and estimated remaining throughput (based on the application's bandwidth requirements). This selection policy also helps in reducing the overload on any single SNAPT. However, there is still a limitation due to the architecture of Internet routing: it is not possible to change the SNAPT handling an application until its data connections end. Figure 2 shows WiOptiMo's architecture for a WMN. The SNAPTs can be managed by private administrators (otherwise called mobility service providers), who may require a fee for the use of their mobility service. This circumstance might foster the competition between mobility service providers, forcing them to increase the quality of provided service and benefit the entire WMN.

B. Implementation changes

We adapted WiOptiMo's implementation (both CNAPT and SNAPT) for low profile devices and to provide a fast handover procedure. Figure 3 shows the changes to the basic implementation of WiOptiMo. A TCP control socket still manages the communication between a CNAPT and a SNAPT. It provides network configuration parameters (e.g., the Maximum Transmission Unit - MTU - of the underlying network) and also transmits data packets in a fall-back mode when middle-boxes, such as firewalls and/or Network Address Translation (NAT), block UDP packets. Further, the control socket is used to authenticate the CNAPT and to exchange a session key for providing data authenticity and integrity during a handover. The CNAPT relays data packets to SNAPTs (and vice versa) using UDP sockets—this solution increases performance during handovers, because UDP does not need to retransmit lost packets nor does it perform any connection setup. When a SNAPT receives a UDP data packet, it validates it using HMAC [24] and tests it against replay attacks using a sequence number. During handovers (i.e., when the source IP address of data packets changes), the SNAPT updates the return IP address for the flow and transmits a keep-alive request to the CNAPT, which will reset the control connection or hasten the detection of a timeout. This event will then trigger the re-establishment of the control socket connection to the SNAPT.

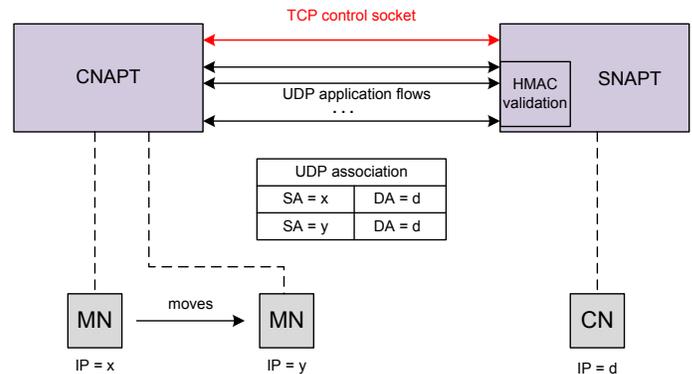


Figure 3. WiOptiMo adaptation for a WMN.

IV. QOS SUPPORT IN WIOPTIMO

In the next sections, we summarize the underlying mechanism and the main experimental results in terms of performance of the flow classification and aggregation scheme in our WiOptiMo framework that we presented in [1].

A. Flow classification

Since WiOptiMo relays each outgoing data flow from a client to a server application (through the link between CNAPT and SNAPT), every flow from a mobile device to its intended destination can be managed separately, according to its characteristics. We exploit this capability to relay data flows to different SNAPTs based on their delay and throughput needs, this way meeting the QoS requirements of applications. In this regard, we identified four different flow classes according to the minimum throughput and maximum delay requirements of applications: *High Throughput and High Delay* (HT & HD),

High Throughput and Low Delay (HT & LD), Low Throughput and High Delay (LT & HD), Low Throughput and Low Delay (LT & LD). Table I presents performance requirements of the most popular applications, along with their correspondent classification. In terms of throughput, the minimum threshold for classifying HT flow classes is 64kbit/s. We set the maximum delay for LD classes to 1s. As previously stated, during the normal workflow, a CNAPT periodically measures delay (one-trip time) and throughput (amount of received data over a time period) towards the different SNAPTs. Then, for each application flow, it detects the class type on the basis of process name, protocol and port number. Every class has an assigned delay and throughput requirements, and data flows get relayed to a SNAPT that meets their delay and throughput requirements.

TABLE I. Applications requirements based on throughput and delay, and their classification.

Application	Class	Min throughput/Max delay
Skype / Video and Voice	HT & LD	128kbit/s / 200ms
Skype only Voice	HT & LD	30kbit/s / 500ms
SSH Client	LT & LD	10kbit/s / 200ms
Web Browser	HT & HD	- / 5s
FTP Client	HT & HD	- / 5s
Google Hangout Video	HT & LD	256kbit/s / 200ms
Google Hangout Chat	LT & LD	10kbit/s / 3s
Remote Desktop Client	LT & LD	- / 200-500ms
Team Viewer	HT & LD	- / 200-500ms
Applets / Widgets	LT & HD	- / 10-30s
Default TCP	LT & HD	- / -
Default UDP	HT & LD	- / -

While our solution for flow classification is conceptually similar to DiffServ [25], it does not have its drawbacks. First, flow classification is performed dynamically per SNAPT, so that new flows are allocated depending on the current network performance statistics (e.g., the increase of the delay with the increase of the load). Second, our framework might refuse to serve a flow if its QoS requirements cannot be met, hence avoiding to disrupt the traffic already allocated. Moreover, the routing layer, as explained in [23], knows which traffic is managed by WiOptiMo. In this way, a QoS-aware routing mechanism can be executed whenever needed. In particular, network statistics about each single flow are reported to the routing layer so that there is no loss of granularity in the traffic management.

V. FLOW AGGREGATION MECHANISM

The class based aggregation technique implemented in our WiOptiMo framework allows to enhance its performance, to efficiently handle applications flows with short frequent sessions (e.g., DNS requests), to optimize wireless link utilization and to increase fairness between competing flows (which is a major drawback when wireless links have high latency [26]). Classified flows that belong to the same class are treated as a single aggregate and transmitted to a SNAPT using the same UDP socket. Our objective is to maximize the utilization of the available link bandwidth and reduce network overhead, thereby increasing the achieved throughput without significantly impacting the latency requirements.

Figure 4 presents the details of our aggregation mechanism. We implemented four connection queues, one for each of the application classes defined in Section IV-A. The queues feed

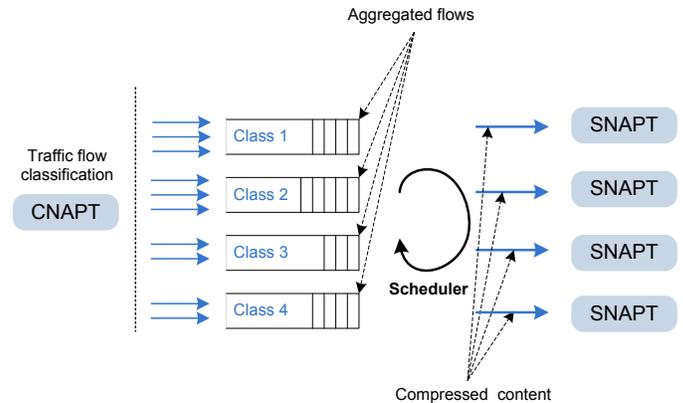


Figure 4. Software architecture of the aggregation scheme.

TABLE II. Different Parameters of the Experiment.

Application Class	Packet Size (Bytes)	Range of bit-rate (bit/s)	Range of Flows
HT & HD	1360	1M - 20M	1 - 5
HT & LD	576	128k - 2M	1 - 5
LT & HD	1360	15k - 1M	1 - 5
LT & LD	100	15k - 128k	1 - 5

HT - High Throughput
 HD - High Delay
 LT - Low Throughput
 LD - Low Delay

into a scheduler, which uses a connection strategy based on flows' priority: the scheduler sends classes with more stringent requirements in terms first of delay and then of bandwidth — this is implemented as a simple static priority queue, cycling through LT & LD, HT & LD, LT & HD and HT & HD queues at dynamic intervals, depending on the processed traffic. To reduce the amount of exchanged data, we enabled compression of the aggregated flows — packets are appended to the aggregated compound until their cumulative compressed size does not exceed the 70% of the underlying network's MTU. We chose this threshold to maximize the effectiveness of aggregation without having to resort to a slower algorithm.

VI. PERFORMANCE OF THE FLOW AGGREGATION SCHEME

We experimentally assessed the performance and QoS support of WiOptiMo with flow aggregation.

A. Performance of WiOptiMo with flow aggregation

To evaluate the performance of our flow aggregation scheme, we conducted experiments in three different scenarios:

- 1) Baseline: without WiOptiMo.
- 2) WiOptiMo basic.
- 3) WiOptiMo with flow aggregation mechanism.

Measurements showed that the performance of the baseline and WiOptiMo basic configurations are comparable (the degradation on throughput and the additional end-to-end delay introduced by the WiOptiMo solution are negligible, as presented also in [10]). For this reason, we report only the results for the baseline and WiOptiMo with flow aggregation scenarios, and show that our flow aggregation scheme achieves a better link utilization and reduces the amount of bytes exchanged in the network.

Our experiment setup was composed by a WiOptiMo SNAPT server and a WiOptiMo CNAPT client (installed on a Dell Precision M4300 with LinkSys Dual-Band Wireless A+G PCI Card), connected through a Netgear WNDR3800 wireless router (with OpenWRT 12.09). To avoid interference with nearby 802.11 access points operating on the 2.4 GHz band, we enabled only 802.11a networking on our router. Both client and server operated on a Linux distribution (Ubuntu 12.04 with Linux kernel 3.11).

We used the *Iperf* [27] network testing tool to send a stream of UDP packets (at a specific bit-rate) to server and measured the number of bytes sent between client and server using the *dumpcap* utility [28]. Instead of using the default UDP packets generated by *Iperf*—all packets contain same data—we configured the *Iperf* utility to generate UDP packets containing random text stored in a file. We performed experiments under the four different classes described in Section IV-A. For each flow class, we fixed the size of data in every UDP packet transmitted by the *Iperf* utility. We repeated experiments 10 times, to get more reliable results. Table II shows the characteristics of every flow generated by *Iperf* to measure the performance of WiOptiMo (for each application class).

We measured the performance of WiOptiMo by varying the number of flows and bit-rate of each flow, and observing their impact on the percentage of bytes saved on the link, due to flow aggregation and compression. The last is calculated by subtracting pre-aggregation (and compression) bytes and post-aggregation (and compression) bytes, and dividing this difference by the pre-aggregation (and compression) bytes. This metric measures the bytes saved in the packet transfer between the client and server with the flow aggregation configuration, compared to the baseline configuration. It captures the energy spent to transfer data to the server. Since WiOptiMo performs flow aggregation and compression, this metric will enable us to measure the amount of energy that could be saved without impacting the QoS of applications.

Figure 5 shows the percentage of bytes saved for applications with high throughput and high delay network requirements. We observe that for bit rates lower than 10Mbit/s, the percentage of bytes saved increases as the number of flows increases. Even for a single application flow, WiOptiMo

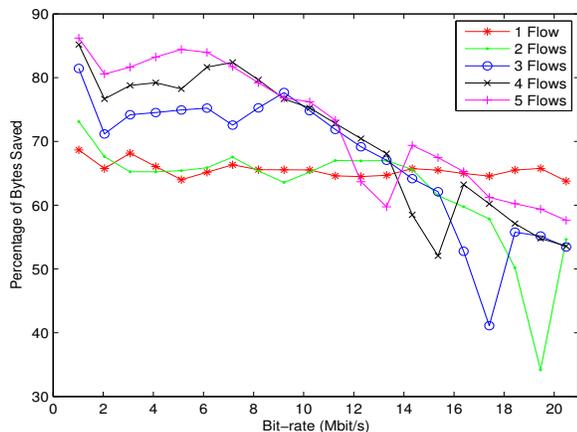


Figure 5. Percentage of bytes saved due to flow aggregation in HT & HD applications.

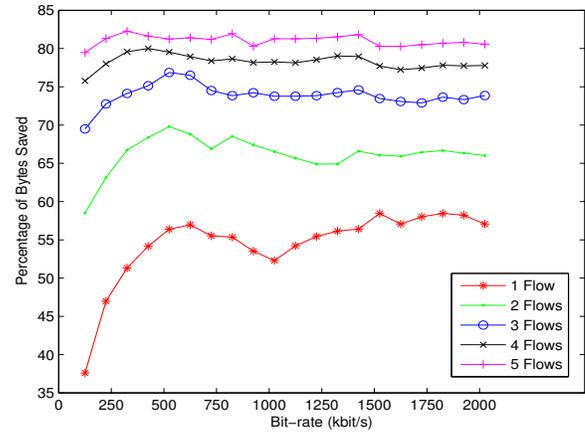


Figure 6. Percentage of bytes saved due to flow aggregation in HT & LD applications.

with flow classification and aggregation helps in reducing, on average, (down) to 60% the amount of data exchanged between client and server. For bit-rates higher than 10Mbit/s, the percentage of bytes saved is still high but its relationship with the number of flows is no longer linear. This behaviour is due to the saturation of the system's modules capacity (wireless card, aggregation and compression mechanisms).

In Figure 6, we observe that when applications have high throughput and low delay requirements, savings by WiOptiMo increase from 38% for single flow to a maximum of 82.5% for applications with 5 flows. For all flows, the percentage of bytes saved increases until the bit-rate reaches about 400kbit/s. For much higher rates we observe that the percentage of bytes saved remains constant.

For low throughput and high delay tolerant applications (see Figure 7), we observe that for low bit-rates (~ 125 kbit/s), the percentage of bytes saved is not significant because no additional savings could be achieved by compressing and aggregating data packets arriving at long intervals of time. For higher bit rates (that is after the size of the aggregated packets allows better compression), savings increase and then

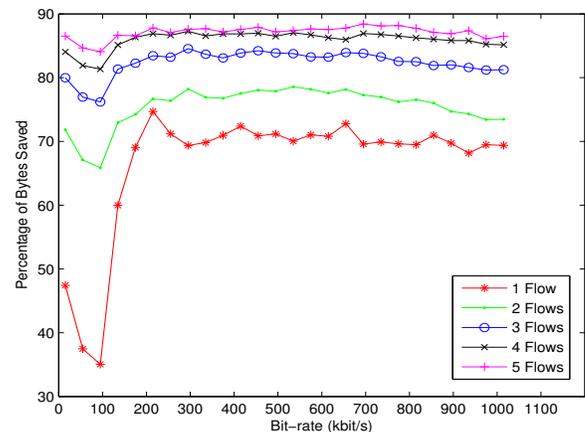


Figure 7. Percentage of bytes saved due to flow aggregation in LT & HD applications.

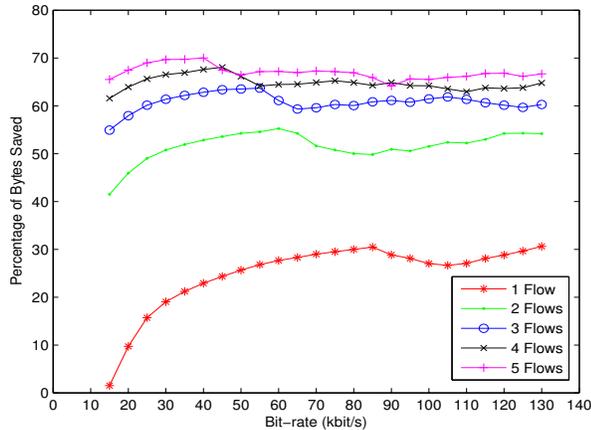


Figure 8. Percentage of bytes saved due to flow aggregation in LT & LD applications.

stay constants (we can achieve a maximum savings of around 90%). In Figure 7, we also observe that savings achieved by WiOptiMo increase as the number of application flows increases.

Finally, for applications with low throughput and low delay requirements, we could achieve a maximum saving of 70% (see Figure 8). Even at very low bit-rate (~ 20 kbit/s), WiOptiMo is able to save 10% of the data transferred between client and server.

B. QoS support by WiOptiMo

To test the capability of the WiOptiMo with an aggregation schema to provide QoS support, we setup a wireless mesh network testbed composed by three static Internet-sharing nodes and two wireless mobile nodes. Each static node consists of an ALIX.2D2 system board, which supports two mini-PCI radios. We used one Wistron DNMA92 miniPCI card for each board, which is in turn connected to two 802.11n antennas. Each board mounts a 500 MHz AMD Geode LX800 processor and 256 MB DDR DRAM, runs Debian Wheezy 7.0 with Linux Kernel 3.12.6, and uses an ath9k driver for WiFi. We used two ASUS EeePC 900 (with an Atheros 5008 Wireless Card, a 900MHz Celeron Processor and 1GB DDR RAM) as mobile nodes in our experiments. They operated on Debian Wheezy 7.0 with an ath5k WiFi driver.

We utilized Iperf and measured the throughput between client and server using two different flow classes (HT & LD and HT & HD), in two distinct configurations: with a single SNAPT and with two SNAPTs.

To complete the hardware setup, we installed WiOptiMo SNAPT on two Dell Optiplex 760 (servers) and a Lenovo ThinkPad T410a had WiOptiMo CNAPT installed on it. Both machines operated on a Linux distribution (Ubuntu 12.04 with Linux kernel 3.11). Two static nodes (gateways, A and C) and two servers were connected to the Internet with an Ethernet connection, while the rest of the nodes (B) participate in the mesh network. We set the bandwidth of Ethernet connection to 10Mbit/s. The gateways performed NAT between the mesh network and the Internet. We ran the Optimised Link State Routing Protocol daemon (OLSRd, version 0.6.2) [29] on each node for network path resolution and configured the network

to ensure that the two SNAPTs could be reached by separate gateways. The final testbed architecture is shown in Figure 9.

Results show that a software configuration with multiple SNAPTs increases the network throughput and then helps preserving the QoS of applications. This is clearly visible in Figure 10, which illustrates the throughput comparison in a single SNAPT and in a double SNAPT (with different network delays and accessible from separate gateways) configuration. In the first scenario, the available bandwidth gets divided equally between the two application classes. In the second scenario, the HT & HD class achieves on average higher throughput compared to HT & LD class because the data of HT & LD class always gets routed to the SNAPT with lowest delay. Specifically, in the two SNAPT scenario, we observe a higher throughput compared to the bandwidth available towards each single gateway. Finally, we did not observe any significant additional delay in the network due to the introduction of WiOptiMo.

VII. MIPv6 COMPARISON

To assess the performance of our WiOptiMo framework with respect to SoA protocols, we compared the behavior of WiOptiMo and MIPv6 [7], which is the standard proposed by IETF to handle mobility of Internet hosts for mobile data communication in IPv6 networks. We focused on IPv6 since it is the basis of the future All-IP networks, as it can be seen for example with the 3GPP decision of adopting IPv6 as the only IP version for an IMS.

The adaptation of our WiOptiMo framework for IPv6 networks was straightforward: the sockets used by WiOptiMo for communications were upgraded to use both IPv4 and IPv6, while a patch, specifically developed for this test, was added to the framework to ensure the exclusive use of the IPv6 protocol. Internal data structures were already designed to store and process IPv6 traffic, so no further modifications were needed.

We measured the following performance parameters:

- 1) *Handoff latency*. Defined as the time interval between the last data segment received through the old path and the first data segment received through the new path from mobile host to corresponding node (CN).
- 2) *Packet loss rate*. Defined as the number of lost packets due to handover divided by the total number of packets sent by the CN.

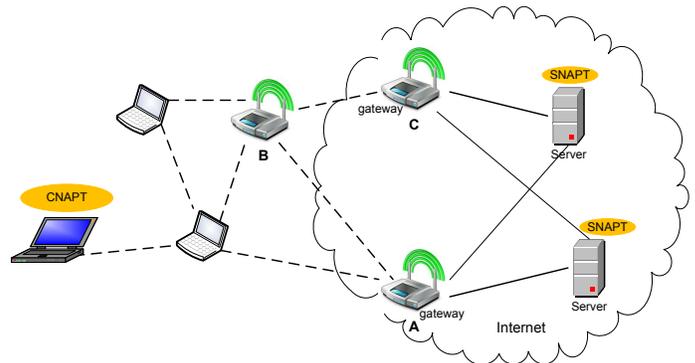


Figure 9. Testbed mesh network architecture.

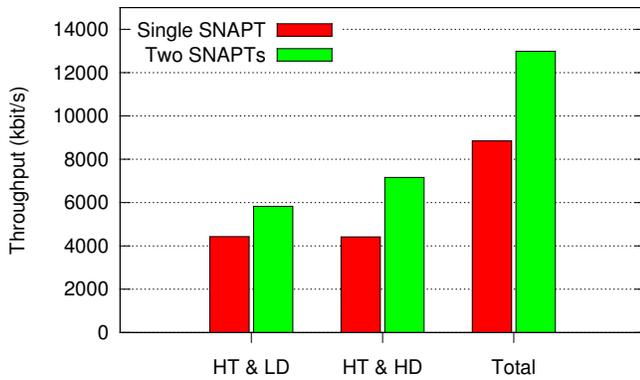


Figure 10. Throughput with multiple SNAPTs.

- 3) *Throughput*. Defined as the total useful bits that can be delivered to the mobile host upper layer application, divided by the time (estimate of the average transmission speed).

The baseline for the performance comparison is a standard IPv6 network configuration, without any mechanism for mobility support.

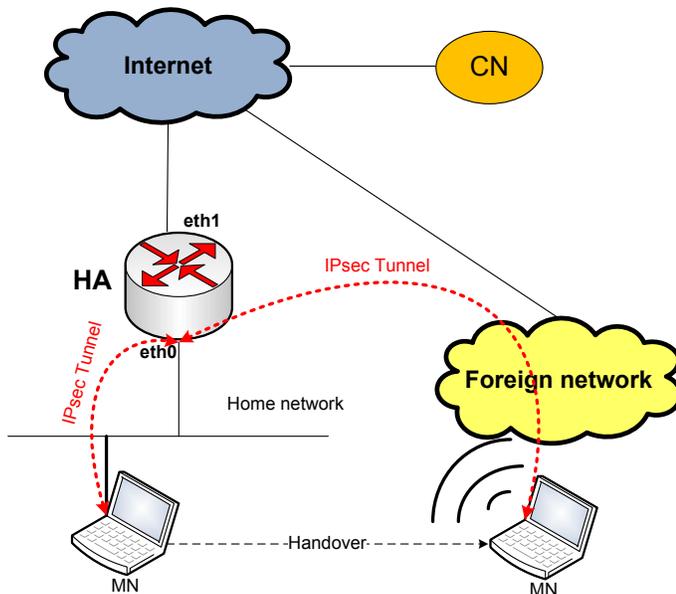


Figure 11. MIPv6 testbed setup

A. MIPv6 Testbed description

We setup a Mobile IPv6 testbed with IPsec static keying, using the UMIP open source implementation for Linux [30]. UMIP supports the IETF RFC RFC6275 (Mobile IPv6)[7]. Figure 11 shows the main elements of the testbed setup, which are:

- The **Home Agent (HA)**. Its egress interface (*eth1*) is connected to the Internet, while its *eth0* interface is connected to the home link of the mobile node.
- The **Mobile Node (MN)**. The MN is initially connected to its Home network using its ethernet interface. Then, after a handover, it connects to a

foreign network using its WiFi interface. The traffic exchanged between the MN and its HA is IPsec protected (tunnel mode).

We installed the HA on a Dell Optiplex 760 computer running a Debian 7.6 (with Linux kernel v. 3.14-2) distribution. The MN we used for our experiments was a HP Folio i3 laptop, running Debian 7.6, equipped with an 802.11b/g/n WiFi card.

In the standard MIPv6 configuration, the communication between the MN and the CN is routed through the HA. To enhance the performance of MIPv6, the Route Optimization (RO) protocol was introduced. The RO enables a MN and a CN to communicate directly, bypassing completely the HA. The RO technique works in this way: after a handover, when the MN receives the first tunneled message from the HA, the MN informs the CN about its new *care-of-address*, by sending a Binding Update (BU) message. The CN stores the home address plus care-of-address into its Binding Cache. Once the new entry is stored, communication directly takes place between MN and CN. To make RO secure, an authentication and encryption mechanism between MN and CN must be set up. The current MIPv6 specification defines that Return Routability (RR) [7] should be used for authentication between MN and CN. The RR procedure assumes that a CN has a private key and a random number that it renews at regular intervals. Although the RR procedure can be easily setup in a laboratory testbed, it is unlikely that every CN is configured for accepting our public certificate. Furthermore, setting up the RR mechanism in every CN is costly. Actually, none of the IPv6-enabled hosts in the 100 Top Internet Websites [31] are configured to be a CN. For this reason, we setup our MIPv6 testbed without RO.

B. WiOptiMo Testbed description

We setup a testbed with WiOptiMo in a single SNAPT configuration scenario. We installed WiOptiMo SNAPT on a Dell Optiplex 760 (server), running a Linux distribution (Ubuntu 14.04). WiOptiMo CNAPT was installed on the HP Folio i3 laptop.

We tested the performance of WiOptiMo in the scenario of WiFi micro-mobility (i.e., handover between WiFi access points of the same provider). We simulated a MN moving between the coverage area of two 802.11 access points with different SSID and IP networks, by manually switching the connection to the access point with the *wpa_gui* tool, a GUI interface for *wpa_supplicant* [32] that enables a user to choose which configured network to connect to.

C. Results

To measure the handover latency, we first connected our MN to an 802.11n access point and then induced an IP and gateway change in the WiFi network, by manually connecting to a new 802.11 access point. As a consequence, the connection was re-routed through a new gateway. To be deterministic, we did not use DHCP to get the IP address of the MN, but used a static IP network configuration. In typical WMNs without any mobility support mechanism, the change of gateway implies the change of external IP address and the need of re-establishing the connection. We used WiOptiMo and MIPv6, and tested their capability to spot the route change and preserve an ongoing transport session.

In WiOptiMo, handover latency is affected by the protocol used by the application. If TCP protocol is used, latency depends on the timeout used by WiOptiMo to detect a broken link or a network traffic stall. Since we wanted to provide absolute values for latency, we measured it by running the standard *ping* utility, with the flood (-f) option. In the flood ping, for every *ECHO_REQUEST* sent a period "." is printed, while for every *ECHO_REPLY* received a backspace is printed. This provides a rapid display of how many packets are being dropped. Since we did not specify any *interval* seconds between sending each packet, packets were transmitted without waiting. Session length of ICMP packets was 60s, packet size was 84 bytes, packets were output as fast as they came back or at one hundred times per second (whichever is more). During the test, we switched the connection between the access points so that the client's ICMP connection was re-routed to a different gateway. At the server side we logged all the socket connections received by SNAPT. We registered the handover latency for WiOptiMo by measuring the time it took for SNAPT to receive ICMP packets (generated by ping at client side) from the new gateway. We iterated this experiment for 50 times.

To measure the handover latency for MIPv6, we transmitted ICMP packets between the MN and the server where we previously run the SNAPT, and computed the difference between the timestamp of the last ICMP packet from the old gateway and the first ICMP packet from the new gateway.

To understand the impact of the overhead introduced by the mechanisms used by WiOptiMo and MIPv6 to manage handover, we recorded the time to complete a layer 3 switch without any mobility support mechanism. This time comprises the following components:

- Disassociation/deauthentication from the current access point.
- Authentication/association to the new access point.
- WPA key negotiation.
- Static IP address loading.

To measure the degradation on throughput and the additional end-to-end delay introduced by WiOptiMo and MIPv6, we used the *netperf* [33] (version 2.6.0) benchmark utility. Netperf is composed of a client and a server (*netserver*) application. The client was installed and run on the MN. It takes as input the IP address of the server, the server port number for TCP control connection and the TCP packet size (bytes). Each experiment lasted for 10 seconds and the netperf client application gave as output the observed throughput (in kbit/s) and the end-to-end delay (in msec). The socket buffer sizes at client (send buffer) and server (receive buffer) were set to their default standard values in Linux. The default TCP send and receive buffer size was 16,384 bytes and 87,380 bytes respectively. The size of each packet transmitted by the client was the same as the send buffer size (i.e., 16,384 bytes). The netserver installed at the server side listened at port number 12865 (default value) for control connections initiated by the client. Degradation on throughput and additional end-to-end delay were measured by running the netperf client application a) without any mobility framework, b) with a MIPv6 setup and c) with WiOptiMo running in background.

Handoff latency: Figure 12 reports the normalized probability density function of the time required by WiOptiMo and MIPv6 to perform the overall handover process when the MN connects to a new access point with a different gateway, while downloading a file via HTTP. The mean of the latency time for WiOptiMo and MIPv6 is shown in Table III. As it can be seen, mean handover latency time is similar for the two solutions, but WiOptiMo slightly outperforms MIPv6. The impact of the mechanisms for managing handover on latency is clearly visible by looking at the mean time required by the operating system for performing a layer 3 switch between the two different access points. It can be observed that mechanisms for managing handover accounts for about 2/3 of the latency time.

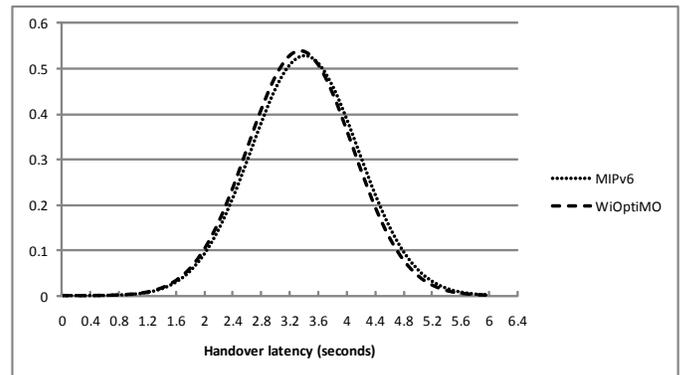


Figure 12. Probability density function of handover latency with no DHCP protocol

TABLE III. Measured Mean Handoff Latency

	Handoff Latency
Without mobility support	1.20 s
WiOptiMo	3.34 s
MIPv6	3.40 s

Throughput: The set of measurements on throughput show that the degradation introduced by WiOptiMo is acceptable and that the throughput experienced in WiOptiMo outperforms (more than six orders of magnitude better) the throughput experienced in MIPv6 (Table IV).

WiOptiMo reduces throughput only by less than 7%: this is mainly due to the computational overhead that is needed for capturing packets at the CNAPT side and signing them, and for checking their integrity at the SNAPT side (and doing the reverse). In standard MIPv6, the drop in throughput is significant (more than 41%) and mainly due to the lack of route optimization: instead of sending packets directly to the MN, the correspondent node sends packets to the MN's home address, which will then encapsulates and forward them to the MN. This mechanism clearly decreases the throughput.

End-to-end delay: To measure end-to-end delay, we used the TCP request/response test of netperf. The request/response performance test consists in executing a transaction, which includes the exchange of a single request and a single response of given sizes. From a transaction rate, the one way and round-trip average latency can be inferred. The TCP request/response

TABLE IV. Measured Throughput

	Throughput
Baseline	50.09 Mbits/s
With WiOptiMo	46.77 Mbits/s
With MIPv6	29.45 Mbits/s

test can be invoked with netperf using the $-t$ option with TCP_RR as argument, and the $-r$ option to set the request and/or response sizes.

Table V reports the results for the end-to-end delay, obtained by running the netperf utility when the TCP packet size varies from 1 byte to 10^8 bytes. The client was connected through a 802.11n network to the netserver server. As it can be seen from measurements, under the same network conditions (default TCP send and receive buffer size set to 16,384 bytes and 87,380 bytes respectively), end-to-end delay depends on the packet size (it increases as the packet size increases). Both WiOptiMo and MIPv6 have worst performances compared to a network configuration without mobility frameworks. WiOptiMo always outperforms (from about 10 to about 3 orders of magnitude) MIPv6. In comparison with the baseline configuration, the performance degradation of MIPv6 is noticeable (up to 13 orders of magnitude) for small packet sizes (1 to 10^4 bytes), while the overhead introduced by WiOptiMo is on average only 1.2 orders of magnitude. Finally, the degradation in percentage on end-to-end delay introduced by WiOptiMo is smaller for large packet sizes. In a typical video streaming usage scenario, which involves large application packet sizes, WiOptiMo has a minor impact on end-to-end delay performance.

VIII. CARMNET USAGE SCENARIO

We demonstrated that WiOptiMo has better performance than MIPv6 in terms of handover latency, packet loss rate and throughput. In this section, we also show its application in other contexts. In particular, we present a scenario where WiOptiMo is employed to support wireless network coverage extension by its integration with a resource allocation framework called CARMNET. Furthermore, we show how the amount of bytes saved thanks to the WiOptiMo aggregation scheme can be taken into account in the computation of a utility-based resource allocation policy.

A. CARMNET architecture

The idea of the CARMNET system was proposed in [4]. CARMNET enables its end users to share their resources, in particular to share the Internet access. The system consists of multiple components deployed both on a client- and server-side. The CARMNET overall system architecture is illustrated in Figure 13.

The DANUMS Loadable Kernel Module (LKM) [34] is an implementation of the Delay Aware Network Utility Maximization (DANUM) model developed for the Linux environment. DANUM is an optimization framework for wireless multi-hop networks that incorporates the delay factor into the computation of network's flows utility connected to the Network Utility Maximization (NUM) model [35]. The subsystem works in the kernel space, which allows for an integration with the network stack necessary to introduce new queuing and scheduling mechanisms. The OLSR daemon, a popular

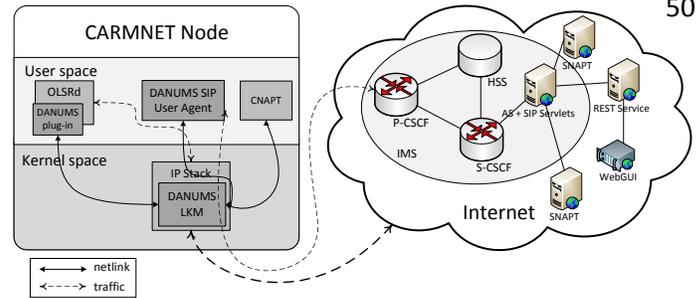


Figure 13. Architecture of the CARMNET system.

implementation of the OLSR routing protocol, is used for the network path resolution and signaling the DANUM-specific information in a distributed way. To ensure a communication between both subsystems, deployed in both the kernel and user space respectively, the Netlink protocol is applied. The Netlink protocol is used to communicate with the client-side part of the CARMNET mobility subsystem based on the WiOptiMo framework. The mobility services are provided by means of the client proxy - CNAPT - installed on the CARMNET wireless node. The role of the component is to intercept traffic flows associated to the mobility service and relay them to the SNAPT server. To ensure scalability and avoid concentrating traffic flows in one single spot, multiple SNAPTs are located on the Internet.

The last subsystem employed on a CARMNET Wireless node is a DANUMS SIP User Agent, which is responsible for exchanging information with an IMS architecture. This integration allows the CARMNET system to use an enhanced IMS infrastructure to provide the session and user management and exploit the unique features of an utility-aware flow control and resource allocation (provided by DANUMS [34]). An user of the CARMNET system can review and/or modify its information via the web application WebGUI integrated through REST service with the IMS infrastructure. One of the goals of the CARMNET system is to make the telecom operator IMS services effectively available to users of WMNs. As a result of integrating the carrier-grade AAA with the NUM-oriented resource management, the system enables the application of utility-based charging based on the *denarii* (i.e., the virtual units of utility) unit of the DANUM subsystem, which may be used as a market-like regulator for utility- and reliability-oriented resource allocation.

The CARMNET system goals include providing an access to the Internet in places without (or with very weak) WiFi signal from the static infrastructure. It may be particularly useful in metropolitan networks, where extending range by means of a static infrastructure can be expensive. In contrast, the CARMNET system, as a distributed and dynamic solution based on the wireless mesh networking paradigm, employs wireless nodes to extend the range of a network.

B. Seamless Horizontal Handover

As presented in [5], there exist at least two scenarios where CARMNET-based solutions can be employed.

The first scenario is based on the network coverage extension concept (see Figure 14). The most of deployed wireless networks incorporate only static infrastructure to provide their services. The infrastructure is very often cheap in maintenance,

TABLE V. End-to-end delay

Packet size [bytes]	1	10	10 ²	10 ³	10 ⁴	10 ⁵	10 ⁶
Delay baseline [milli sec]	0.641	0.653	0.662	0.803	2.112	14.490	134.804
Delay with WiOptimo [milli sec]	0.794	0.799	0.804	0.891	2.331	16.257	156.235
Delay with MIPv6 [milli sec]	8.18	8.235	8.87	10.246	14.02	54.919	407.381
WiOptiMo Degradation [%]	19.27	18.24	19.29	11.5	9.42	10.87	13.71
MIPv6 Degradation [%]	92.16	92.06	92.53	92.15	84.94	73.62	66.9

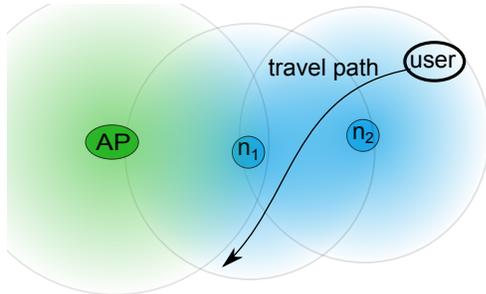


Figure 14. Example network topology in network coverage extension scenario.

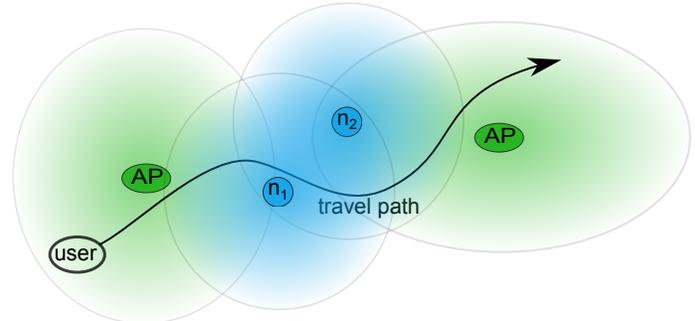


Figure 15. Network topology in handover scenario.

however, it covers only a small area and an extension of its range is expensive. The proposed scenario allows to reduce this cost by shifting it to the users of the network, as the result of adopting the WMN paradigm. This approach benefits from the distributed locations of users - who are rarely concentrated in one place - that leads to quite large overall coverage. To put it into practice, each user device must run the CARMNET client software, which enables sharing Internet access with other users. As described in the previous section, the client software is not only responsible for sharing a network access, but also introduces a resource management system that takes care of the users' willingness to serve their own or other users' traffic.

However, an important issue is how to encourage users to share resources on their devices (e.g., CPU load, battery, bandwidth) and a network access. In this respect, CARMNET defines a virtual currency based on a utility function derived from the NUM optimization problem. The utility function depends on traffic parameters like throughput and delay. In the CARMNET system, the *denarii* virtual currency is utilized to charge users. Furthermore, it enables incorporating an incentive model which may enforce collaboration between users to effectively use the network.

A more complex scenario than the network range extension is the provision of the Internet access by multiple independent infrastructures and wireless networks in a single broad area (see Figure 15). In this scenario, a CARMNET infrastructure can be set up in order to bridge the gap between two (or more) access points. As most users are mobile and use Internet on the move (e.g., they perform a videocall or download some documents), a transition between independent networks can be an issue. In order to solve this problem, there is a need to provide handover services between CARMNET-based networks and the independent wireless networks, which will enable the possibility to transparently transfer ongoing sessions between networks without interruption. In the CARMNET system, the role of the handover service provider is played by the WiOptiMo framework, which is integrated with the rest of the components.

C. The Usage of Virtual Currency

Incentive mechanisms play an important role for encouraging users to use CARMNET-based networks. An incentive mechanism can be defined based on the concept of virtual currency introduced and used by the DANUM framework [34], which was primarily applied only for flow management purposes. The DANUM framework, as the application of the Delay-Aware variant of the Network Utility Maximization framework, determines utility of each served flow according to the flows' parameters like delay and throughput. Then, the value of the virtual currency rate y_f of flow f is calculated as a solution of the primal DANUM problem:

$$\max_{y_f} \sum_f U_f(y_f), \quad (1)$$

subject to the constraints associated with the system of virtual queues (see [34] for detailed description of the Delay-Aware NUM model). The $U_f(y_f)$ function is an utility function defined to represent characteristics of the flow f , i.e., according to the type of service. Examples of utility functions for TCP and UDP protocols are defined in [34].

As a main step of the DANUM flow control mechanism, a virtual unit value is calculated as derivative of the flow's utility. This value is then used to build virtual queues (managed in parallel with the real packet queues). The levels of virtual queues are thereafter used to schedule flows [34].

In [5], a model of the rewarding mechanism was proposed, however, it has been provided only for traffic forwarded inside each CARMNET network. In the scenario of seamless handover between multiple networks (based on WiOptiMo), such a simplified model is not sufficient. As described in Section V, WiOptiMo aggregates multiple flows belonging to the same class into one UDP flow. This approach optimizes the utilization of bandwidth, however, it compromises the capability of the DANUM subsystem to charge users, since flows are aggregated and DANUM is not able to compute the accurate number of virtual units.

Because of the incorrect utility estimation, the CARMNET users' account balance can be charged in a wrong way. For this reason, we propose a modification to the existing charging algorithm, which involves extending the functionality of the SNAPT component by adding an additional reporting mechanism. Since the SNAPT knows exactly the delay and throughput of each aggregated flow, it reports this information to the CARMNET IMS subsystem, which in turn uses it to recalculate the users' traffic reports and the account balance.

At the source node of each flow, the price of the virtual unit is computed using the standard partial derivative formula as follows:

$$\begin{aligned} U'_f(y_f(t)) &= \frac{\partial U}{\partial y}(y_f(t), \mathbf{v}) \\ &= \frac{\partial U}{\partial x}(x(y_f(t), \mathbf{v}), d(y_f(t), \mathbf{v})) \frac{\partial x}{\partial y}(y_f(t), \mathbf{v}) \\ &\quad + \frac{\partial U}{\partial d}(x(y_f(t), \mathbf{v}), d(y_f(t), \mathbf{v})) \frac{\partial d}{\partial y}(y_f(t), \mathbf{v}), \end{aligned} \quad (2)$$

where x and d denotes the end-to-end throughput and delay, respectively, and the exact values of utility derivatives are calculated using the linear interpolation. The values of the derivatives of x and d with respect to y are assumed to be constant [34] and set experimentally. The vector \mathbf{v} represents other than delay and throughput flow parameters, which may affect user-perceived utility for a given flow, e.g., the jitter or packet loss. Formula 2 is used for offline price recalculations (according to reported values of delay and throughput) of served and aggregated flows regarding the real utility. In the next step, the new price is used to update users' account balance to the appropriate value.

In other words, although seamless handover provided by WiOptiMo is an important feature, it is much harder to correctly estimate flow utility in the mobility scenario. This might lead to the situation where the DANUM resource management system under- or overestimates the utility of traffic and, as a result, to the decrease of the service quality. To address this issue, we have introduced a rewarding mechanism for CARMNET users who utilize the WiOptiMo mobility service. This mechanism involves a discount in the virtual currency that is proportional to the amount of traffic served by WiOptiMo (reported to the IMS subsystem by SNAPT). This way, we acknowledge the benefit for the CARMNET network derived from the amount of bytes saved thanks to the use of the WiOptiMo aggregation scheme.

IX. CONCLUSION

In this paper, we have addressed the issue of supporting QoS expectations of mobile users in a wireless mesh networking environment and proposed a flow classification and aggregation scheme based on the WiOptiMo framework, to manage multiple applications with different QoS requirements. We evaluated our scheme on a Linux-based wireless mesh network testbed and showed that the aggregation mechanism improves network performance in terms of link utilization and QoS, while still providing mobility support. We also tested WiOptiMo in a IPv6 network and demonstrated that it outperforms the Mobile IPv6 protocol in terms of handover latency, packet loss rate and throughput. Finally, we have proposed an incentive mechanism for motivating nodes that

utilize the WiOptiMo mobility framework to share their network resources with other nodes of a wireless mesh network. The strengths of our framework are that it does not require any changes to be made to the network protocol stacks of either the mobile or fixed end systems, it does not suffer from the scalability problems of Mobile IPv6 because it enables an efficient management of local mobility, and it can be easily integrated into a utility-based resource allocation framework.

ACKNOWLEDGMENT

This work is supported by a grant from Switzerland through the Swiss Contribution to the enlarged European Union (PSPB-146/2010, CARMNET).

REFERENCES

- [1] D. Gallucci, S. Mudda, S. Vanini, and R. Szalski, "A Flow Aggregation Scheme for Seamless QoS Mobility Support in Wireless Mesh Networks," in *MOBILITY 2014, The Fourth International Conference on Mobile Services, Resources, and Users*, Paris, France, July 2014, pp. 32–37.
- [2] S. Jakubczak, D. G. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan, "Link-alike: using wireless to share network resources in a neighborhood," vol. 12, no. 4. ACM, 2009, pp. 1–14.
- [3] C. Middleton and A. B. Potter, "Is it good to share? A case study of FON and Meraki approaches to broadband provision," in *Proceedings of International Telecommunications Society 17th Biennial Conference*, Montreal, 2008.
- [4] M. Glabowski and A. Szwabe, "Carrier-Grade Internet Access Sharing in Wireless Mesh Networks: the Vision of the CARMNET Project," in *AICT 2013, The Ninth Advanced International Conference on Telecommunications*, June 2013, pp. 113–116.
- [5] P. Walkowiak, R. Szalski, S. Vanini, and A. Walt, "Integrating CARMNET System with Public Wireless Networks," *ICN 2014, The Thirteenth International Conference on Networks*, pp. 172–177, February 2014.
- [6] C. Perkins, "IP Mobility Support for IPv4, Revised," RFC 5944 (Proposed Standard), Internet Engineering Task Force, 2010, last retrieved: 06.07.2015. [Online]. Available: <http://www.ietf.org/rfc/rfc5944.txt>
- [7] C. Perkins, D. Johnson, and J. Arkko, "Mobility Support in IPv6," RFC 6275 (Proposed Standard), Internet Engineering Task Force, Jul. 2011, last retrieved: 06.07.2015. [Online]. Available: <http://www.ietf.org/rfc/rfc6275.txt>
- [8] R. Koodli, "Mobile IPv6 Fast Handovers," RFC 5568 (Proposed Standard), Internet Engineering Task Force, 2009, updated by RFC 7411. Last retrieved: 06.07.2015. [Online]. Available: <http://www.ietf.org/rfc/rfc5568.txt>
- [9] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier, "Hierarchical Mobile IPv6 (HMIPv6) Mobility Management," RFC 5380 (Proposed Standard), Internet Engineering Task Force, Oct. 2008, last retrieved: 06.07.2015. [Online]. Available: <http://www.ietf.org/rfc/rfc5380.txt>
- [10] G. A. Di Caro *et al.*, "Deployable Application Layer Solution for Seamless Mobility Across Heterogeneous Networks." *Ad Hoc & Sensor Wireless Networks*, vol. 4, no. 1-2, pp. 1–42, 2007.
- [11] T. Ernst, "Network Mobility Support Goals and Requirements," RFC 4886 (Informational), Internet Engineering Task Force, 2007, last retrieved: 06.07.2015. [Online]. Available: <http://www.ietf.org/rfc/rfc4886.txt>
- [12] K. Ahmavaara, H. Haverinen, and R. Pichna, "Interworking architecture between 3GPP and WLAN systems," *Communications Magazine, IEEE*, vol. 41, no. 11, pp. 74–81, 2003.
- [13] 3GPP Specifications. Last retrieved: 06.07.2015. [Online]. Available: <http://www.3gpp.org/specifications/>
- [14] M. Buddhikot, A. Hari, K. Singh, and S. Miller, "MobileNAT: A New Technique for Mobility Across Heterogeneous Address Spaces," *ACM Mobile Networks and Applications*, vol. 10, no. 3, pp. 289–302, 2005.
- [15] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, 2005, pp. 675–684.

- [16] V. Navda, A. Kashyap, and S. R. Das, "Design and Evaluation of iMesh: an Infrastructure-mode Wireless Mesh Network," in *World of Wireless Mobile and Multimedia Networks. WoWMoM 2005, Italy, June 2005*, pp. 164–170.
- [17] Y. He and D. Perkins, "BASH: A Backhaul-Aided Seamless Handoff Scheme for Wireless Mesh Networks," in *World of Wireless, Mobile and Multimedia Networks. WoWMoM 2008*. IEEE, June 2008, pp. 1–8.
- [18] R. Huang, C. Zhang, and Y. Fang, "A Mobility Management Scheme for Wireless Mesh Networks," in *Global Telecommunications Conference. GLOBECOM'07. IEEE*, Washington DC, USA, November 2007, pp. 5092–5096.
- [19] M. Sabeur, G. Al Sukkar, B. Jouaber, D. Zeghlache, and H. Afifi, "Mobile Party: A Mobility Management Solution for Wireless Mesh Network," in *Wireless and Mobile Computing, Networking and Communications. WiMOB 2007*, October 2007, p. 45.
- [20] S. Speicher and C. H. Cap, "Fast Layer 3 Handoffs in AODV-Based IEEE 802.11 Wireless Mesh Networks," in *Wireless Communication Systems. ISWCS'06*. IEEE, 2006, pp. 233–237.
- [21] Y. Amir, C. Danilov, R. Musuăloiu-Elefteri, and N. Rivera, "The SMesh Wireless Mesh Network," *ACM Transactions on Computer Systems (TOCS)*, vol. 28, no. 3, pp. 6:1–6:49, September 2010.
- [22] D. Gallucci, S. Giordano, D. Puccinelli, N. Tejaws, and S. Vanini, "Fixed Mobile Convergence: The Quest for Seamless Mobility," in *Fixed/Mobile Convergence Handbook*. CRC Press, 2010, pp. 185–196.
- [23] S. Vanini, D. Gallucci, S. Giordano, and A. Szwabe, "A Delay-aware NUM-driven Framework with Terminal-based Mobility Support for Heterogeneous Wireless Multi-hop Networks," in *IEICE Information and Communication Technology Forum 2013*, 2013.
- [24] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104 (Informational), Internet Engineering Task Force, Feb. 1997, updated by RFC 6151. Last retrieved: 06.07.2015. [Online]. Available: <http://www.ietf.org/rfc/rfc2104.txt>
- [25] IETF DiffServ Working Group page. Last retrieved: 06.07.2015. [Online]. Available: <http://datatracker.ietf.org/wg/diffserv/charter>
- [26] R. Chakravorty, S. Katti, J. Crowcroft, and I. Pratt, "Flow Aggregation for Enhanced TCP over Wide-Area Wireless," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003, pp. 1754–1764.
- [27] iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool. Last retrieved: 06.07.2015. [Online]. Available: <https://github.com/esnet/iperf>
- [28] dumpcap - The Wireshark Network Analyzer. Last retrieved: 06.07.2015. [Online]. Available: <http://www.wireshark.org/docs/man-pages/dumpcap.html>
- [29] An ad-hoc wireless mesh routing daemon. Last retrieved: 06.07.2015. [Online]. Available: <http://www.olsr.org>
- [30] UMIP - Mobile IPv6 and NEMO for Linux. Last retrieved: 06.07.2015. [Online]. Available: <http://www.umip.org/>
- [31] List of most popular websites. Last retrieved: 06.07.2015. [Online]. Available: http://en.wikipedia.org/wiki/List_of_most_popular_websites
- [32] Linux WPA/WPA2/IEEE 802.1X Supplicant. Last retrieved: 06.07.2015. [Online]. Available: http://w1.fi/wpa_supplicant/
- [33] Netperf Homepage. Last retrieved: 06.07.2015. [Online]. Available: <http://www.netperf.org/netperf/>
- [34] A. Szwabe, P. Misiorek, and P. Walkowiak, "Delay-Aware NUM System for Wireless Multi-hop Networks," in *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European*, Vienna, Austria, April 2011, pp. 530–537.
- [35] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

Evaluation of an Uncertainty Aware Hybrid Clock Synchronisation System for Wireless Sensor Networks

Christoph Steup, Sebastian Zug and Jörg Kaiser

Department of Distributed Systems

Faculty of Computer Science

Otto-von-Guericke-University

Magdeburg, Germany

Email: {steup,zug,kaiser}@ivs.cs.ovgu.de

Abstract—Wireless Sensor Networks, aiming to monitor the real world’s phenomena reliably, need to combine and post-process the detected individual events. This is not possible without reliable context information for each event. One important aspect of this context is time. It enables ordering of events as well as deduction of further data like rates and durations. Consequently, an unreliable time base affects all aspects of further processing. Any uncertain time information generates uncertain values and decisions, which jeopardize the correct behaviour of the system unless the system knows them. Therefore, the synchronisation of the clocks of the individual nodes is of high importance to the reliability of the system. On the other hand, tight and reliable synchronisation typically induces a large message overhead, which is often not tolerable in WSN scenarios. This paper evaluates a new hybrid synchronisation mechanism enabling tight synchronisation in single-hop environments and looser synchronisation in multi-hop environments. The lack of a guaranteed synchronisation precision is mitigated by an explicit synchronisation uncertainty, which is passed to the application. This enables the application to react to changes in the current synchronisation precision. The new approach is evaluated using the network Simulator OMNET++ and a small scale wireless sensor network to verify the expected performance and assumptions. The new method showed excellent performance in single-hop environments and a decreasing synchronisation precision based on the topological distance in multi-hop scenarios, which are useful results for Wireless Sensor Networks.

Keywords—Wireless Sensor Networks; Clock Synchronisation; Uncertainty

I. INTRODUCTION

Time is a relevant dimension in all aspects of the physical world. Consequently, measuring, comparing and computing time is a very important capability of any technical system observing or interacting with the real world. The approach presented by Steup et al. [1] proposed a method to include a low-overhead uncertainty aware clock synchronisation mechanism for hierarchical networks. The proposed method provides tight synchronisation in single-hop environments and looser synchronisation with a decreasing precision based on the topological distance. In this paper this approach will be explained, compared to existing approaches and evaluated using a simulated environment and a small-scale Wireless Sensor Network (WSN).

WSN gain increasing attention by researchers as well as industry and governments. They provide the ability to monitor

large areas for events efficiently and with small effort. One example is the SafeCast Project [2], which aims to provide people with the ability to cheaply monitor radiation in their vicinity and share this data with others. The forest fire detection system described by Yu et al. [3] is another example aiming to protect people and nature using real-time information fed by a WSN. A third example is the Avalanche system of Michahelles et al. [4] enabling easier detection and rescue of people caught in an avalanche using medical sensor attached to the person. There are many more applications of WSN using real-time data streams to react to environmental conditions, hazards or catastrophes.

As described by Römer and Mattern [5] WSN come in the different forms. The design space consists of multiple dimensions, which contain network topology (from single-hop to graph), granularity (high-power to smart dust), coverage (dense, sparse and redundant), heterogeneity (heterogeneous or homogeneous), communication interface and mobility (from static to high speed mobile). There are also commonalities that all WSN share like perceiving the environment using equipped sensors and a limitation in energy, processing power and network bandwidth. Generally WSN are considered to be disseminating data from the sensor nodes to the sinks. Depending on the application and the limitations of the WSN it is often necessary to handle evaluation or decision making of the data in the network itself. Some WSN like the above described systems need extend this basic dissemination. They either need to deliver decisions to sinks within the network or need to process the data within the network to limit the needed bandwidth. These systems need robust decision making and processing mechanisms to be accepted by people, since a missed warning may cause fatalities, whereas too many wrong warnings lead people to ignore them. Therefore, reliable post processing and context detection are crucial to provide a robust output. In this context time together with space are the most important context attributes a WSN needs to deliver to enable evaluation and decision making. Consequently, the quality of the time base is an important aspect governing the reliability of the output, since an unreliable time base influences the ordering as well as the deduction of events, which in consequence become unreliable as well. Therefore, a reliable, precise global time base is a must-have for each WSN detecting safety relevant events.

The description of the approach starts with a short introduction to basic terms and concepts of clock synchronisation in general in Section II. Afterwards we discuss some relevant existing clock synchronisation protocols in Section III, followed by the description of our concept in Section IV. In Section V, we describe our implementation within the Omnet++ network simulator and our real world evaluation, the tests carried out and their results. The paper closes with a conclusion of our results and some ideas on future work in Section VI.

II. FOUNDATIONS OF CLOCK SYNCHRONISATION

The time base of any digital system is provided by a clock. The clock itself can be modeled as linearly increasing counter, with a defined time period between the clock events. This period is called *granularity* (g) of the clock [6]. Since a clock is a sensor measuring time, it has an *accuracy* (α) describing its difference to the real value of time and a *precision* (π) describing the maximum difference between any two clocks in a system [6]. The accuracy of a clock directly defines the precision of this clock to be: $\pi = 2\alpha$ [6].

Precision and accuracy are typically unbounded for unsynchronized clocks. Therefore, these parameters can only be evaluated after the clocks of a system are synchronized. To be able to evaluate the quality of a single clock other parameters need to be considered. The most important parameter is the *drift* (ρ) [6]. It describes the maximum deviation of the frequency of the clock towards and ideal clock. The frequency deviation between two real clocks is measured by the *skew*. Over time the drift will increase the *offset* (δ) between the real clock and the ideal clock [6]. The skew will increase the offset between any pair of clocks. Consequently, the aim of any synchronisation mechanism is to limit the offset to a certain value defining either precision or accuracy of the clocks of the system. Since drift and clock skew will increase the offset again after a successful synchronisation, the synchronisation needs to be repeated periodically. This *resynchronisation interval* is a very important parameter for most synchronisation protocols as it defines the achievable precision or accuracy [6]. Ideally, the resynchronisation interval depends only on the drift or clock skew, but in real system acquiring, distributing and comparing time stamps takes time itself. This time is another important parameter for synchronisation protocols and is determined by the *critical path*. Estimating the length of this path or minimizing it are two general approaches to increase synchronisation precision or accuracy.

A. Internal and External Synchronisation

The easiest solutions for an accurate, precise and reliable time base are external time sources. One exemplary source is the DCF77 [7] standard used in Germany to distribute the time of the atomic clock in Braunschweig. This standard uses a very simple protocol transmitted over a 77.5 kHz wave to supply whole Germany with only a single sender.

Another possible source is the Global Positioning System (GPS) as described by Dana [8]. GPS is a Time-of-flight based positioning system, which needs a very accurate time reference to infer the time between transmission and reception of the signal. For this purpose one satellite is typically used as a time reference, whereas at minimum three others are used to derive the position. Consequently, GPS always provides a time reference additionally to the position.

GPS and DCF77 are examples of external synchronisation, where a high accuracy reference source is used to synchronize all nodes. This approach can guarantee precision and accuracy based on the used communication and quality of the reference source [9].

An alternative approach is internal synchronisation, which aims to provide a bounded precision without accuracy [9]. This is done by synchronizing the nodes with each other without any external reference. Typical protocols providing internal synchronisation are the Network Time Protocol (NTP) [10] and the Precision Time Protocol (PTP) [11].

Using additional hardware to supply each node of a WSN with an accurate time is often not feasible because of limitations to cost and energy consumption the nodes need to adhere to. Therefore, pure external synchronisation is seldom used. Internal synchronisation protocols like NTP and PTP are too expensive considering message count and are typically not robust enough for the unreliable wireless links. As a consequence, specially adopted time synchronisation protocols for WSN were developed. These protocols can be divided in two groups: Averaging and Non-Averaging protocols. The averaging protocols exchange time stamps between nodes without any hierarchy. Afterwards, they try to achieve consensus on the resulting time by averaging the timestamps [9]. Non-averaging protocols typically have a master distributing its time in the network with slaves adopting this time to mitigate the offset between pairs of nodes [9]. Obviously, the averaging approaches have problems tolerating malicious or faulty clocks, since clock values with large offsets will strongly influence the average. On the other hand, non-averaging protocols using master nodes have problems tolerating a failure of their master node.

In general, all approaches try to provide a trade-off between message overhead and synchronisation precisions. Additionally, they try to tolerate message losses and changes in the topology of the network. However, most of the existing protocols either try to provide a tight synchronisation in a single hop environment and degrade heavily in multi-hop environments or provide a generally looser synchronisation in both. Unfortunately, none of the existing protocols provide the application with information on the current status of the synchronisation, which might be degraded by errors in communication or heavy changes in topology. The next section will discuss some existing clock synchronisation protocols and evaluate their fitness towards a hybrid uncertainty aware clock synchronisation for WSN.

III. STATE OF THE ART

In order to assess the current state of clock synchronisation for WSN, we describe six approaches representing basic concepts in the following section.

A. Reference Broadcast Synchronisation (RBS)

Reference Broadcast Synchronisation as described by Elson et al. [12] is an averaging internal synchronisation mechanism exploiting a physical broadcast in a shared medium. The synchronisation starts with one node transmitting a *NOW*-message to all other nodes. This message serves as an indication for all nodes to take a local time stamp. Afterwards, the timestamps are exchanged between all nodes. This mechanism

reduces the critical path to the transmission time of the *NOW*-message and the local processing time on each node until the local time stamp is taken. This provides very tight synchronisation in single-hop scenarios as long as the computation time is bounded. However, in worst case for n nodes $\mathcal{O}(n^2)$ messages are needed for a single synchronisation round, as visible in Figure 1.

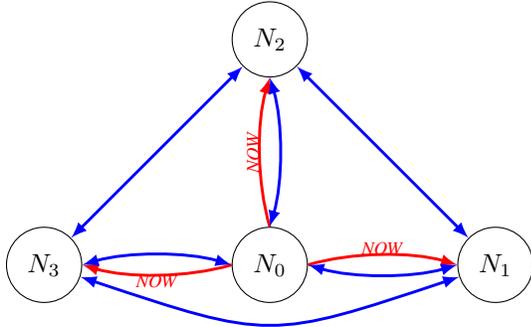


Figure 1. Overview of a single synchronisation round of RBS. The red arrows indicate the *NOW*-message, whereas the blue arrows indicate the messages containing the timestamps of the nodes.

Finally, all nodes individually compute their offset towards the mean of all exchanged timestamps. Towards this end, each node n_i computes its phase offset to each other node n_j over all received broadcasts by another node n_k based on (1). Additionally, a linear regression analyses based on the estimated offsets is used to estimate the clock skew of each node towards the global clock. Using this information all receivers agree on a uniform clock rate and time within the single-hop environment.

$$\delta_{i,j} = \frac{1}{m} \sum_{k=1}^m (ts_{j,k} - ts_{i,k}), i \neq j \neq k \quad (1)$$

In summary, RBS provides tight synchronisation bounds for single-hop environments. However, it reacts very sensitive to mobile or faulty nodes. This is caused by the used averaging mechanism of the protocol. The contribution of all nodes to the averaged new time and clock rate may create large shifts of agreed global clock whenever one node's clock is far off. This is especially problematic whenever a node is only a temporary member of the broadcast group.

B. CesiumSpray (CS)

CesiumSpray by Verissimo et al. [13] is a pseudo-hierarchical non-averaging hybrid clock synchronisation mechanism providing strong failure resilience in real-time networks. The baseline idea of this system is the synchronisation of groups of nodes within a single-hop environment towards an external reference. Consequently, the approach uses internal and external synchronisation mechanism. The authors proposed GPS receivers as the external reference, but other time sources like DFC77 receivers or even precise local clocks are possible. An example network employing CesiumSpray is depicted in Figure 2. As visible, the GPS satellites together with the GPS receivers in the single-hop environments act as a multi-hop backbone to sync the individual clusters.

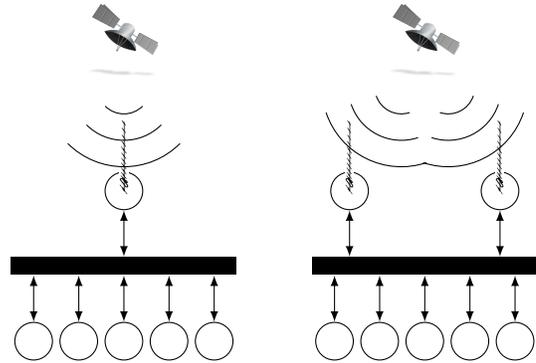


Figure 2. The pseudo-hierarchical structure of an example CesiumSpray clock synchronisation network.

Within each cluster, so called tight broadcasts are used to spray the GPS reference to the other nodes. Special care was taken by the authors to handle faulty GPS references or faulty GPS receiver nodes. This is achieved by acknowledging the broadcast of each GPS receiver node from all other nodes to enable a homogeneous view of the whole cluster. After the reception of a tight broadcast from all local GPS receiver nodes, an agreement is necessary to select a single broadcast. Afterwards, all local nodes sync against the chosen tight broadcast. In case no GPS reference is available, a purely internal synchronisation mechanism is used to preserve precision and accuracy.

The precision of the approach is based on the accuracy of the GPS receivers, which typically is better than $110ns$, and the precision of the internal synchronisation mechanisms. The authors tested their approach in a real-time network using very old Motorola 68020 CPUs and a token-bus network. As a result they achieved a precision of $500\mu s$ with a resynchronisation interval of $150s$.

CesiumSpray is a resilient multi-hop clock synchronisation system for distributed real-time systems. It provides excellent failure resilience with a proven upper bound on the synchronisations precision and accuracy. The precision is dependent on the tightness of the network and the real-time capabilities of the platform used. The drawbacks of the approach consist in the need for real-time capable networks and operating systems to use the strong failure resilience and the need for an external time source providing the "multi-hop" capability. Due to the provided failure tolerance the synchronisation costs are quite high.

C. Delay Measurement Time Synchronisation Protocol (DMTS)

The Delay Measurement Time Synchronisation Protocol described by Ping [14] modifies RBS, see Section III-A, by exploiting low-level hardware access and a non-averaging computation. It extends the *NOW*-message with a time stamp ts_0 taken and inserted just before sending. An averaging is not used anymore, since the transmitted time stamp ts_0 can directly be used by each receiving node. Therefore, the exchange of the individual local time stamps is omitted and the message count is heavily reduced, as visible in Figure 3.

The time between the transmission time ts_M and the reception time ts_S is composed of the interrupt service time

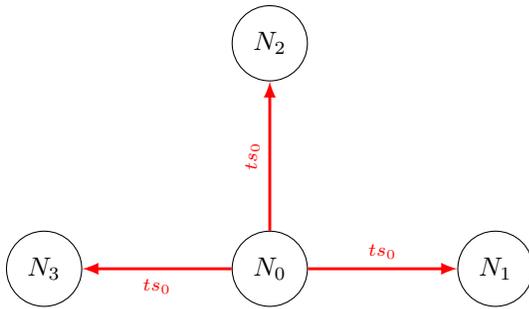


Figure 3. Overview of a single synchronisation round of DMTS. The red arrows indicate the time stamp messages distributed by the master node N_0 .

and the propagation time of the network. To reach similar synchronisation precision as RBS, the author estimates the duration of the interrupt service t_{comp} of the receiving node analytically and modifies the inserted time stamp just before transmission, as shown in Figure 4. If the analysis is correct only the transmission time τ on the medium remains as the critical path.

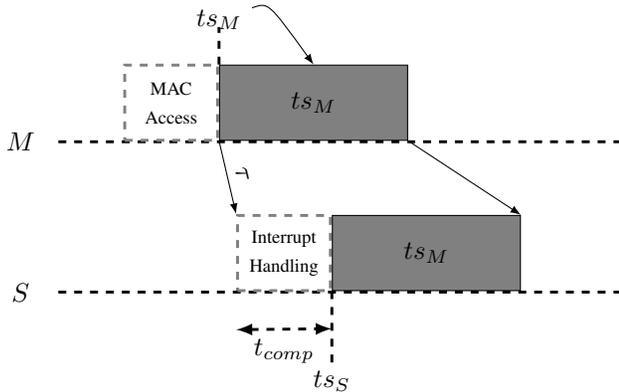


Figure 4. Minimization of the critical path of DMTS.

For this approach to be feasible, a master election is necessary, since a single node needs to transmit its time stamp for multiple rounds until synchronisation in all nodes is reached. After that, the time stamp providing node can freely be chosen from the set of synchronized nodes.

The multi-hop synchronisation of DMTS is based on a hierarchical distribution of the master's time. Each node in the single-hop neighbourhood synchronized with the master will act itself as the master for its own single-hop neighbourhood using its synchronized time. In consequence, the time of the master will distribute over the whole network over time.

The hierarchical structure is non-deterministic, since nodes will always sync themselves to the first node broadcasting a time beacon they hear. Typically, this will create shortest routes from the time master to the individual nodes, which may decrease the reliability of the link and therefore lower robustness.

DMTS provides a high precision clock synchronisation in single-hop neighbourhoods as well as multi-hop synchronisation with slowly degrading performance. It solves the large amount of message necessary for a single synchronisation of

RBS. However, in-depth knowledge of the needed hardware and communication mechanisms as well as low-level hardware access is needed to use it. The protocol has very limited robustness as only a single time stamp is communicated from master to its neighbourhood, which leaves the protocol open to omission failures and faulty nodes spreading wrong clock values when elected.

D. Continuous Clock Synchronisation in Wireless Real-time Applications (CCS)

The Continuous Clock Synchronisation for Wireless Real-time Applications by Mock et al. [15] is a non-averaging master-slave synchronisation method extending the basic clock synchronisation of the 802.11 standard [16]. It is itself the adoption of the approach of Gergeleit and Streich [17] towards 802.11 networks.

In contrast to the standard time synchronisation mechanism of the 802.11 protocol, the clocks of the slaves are not simply set to the time stamp of the master, but are gradually adapted by adjusting their rate. Consequently, the authors propose the concept of virtual clocks (VC) to enable dynamic adjustment of frequency and offset. The behaviour of such a virtual clock is depicted in Figure 5.

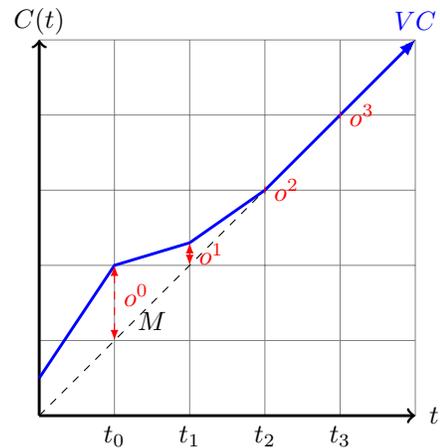


Figure 5. A virtual clock (VC) of a time slave adapting its frequency to compensate offset δ and clock skew towards its master clock M .

Additionally, the precision of the synchronisation is enhanced by dividing the time beacon in a *NOW*-message indicating surrounding nodes a time stamp ts_i needs to be taken and an additional message containing the time stamp ts_m of the *NOW*-message. The major benefit of this approach is the ability to exactly estimate the time stamp of the *NOW*-message minimizing the critical path and omitting any estimation of media access, transmission or computation delays.

The additional message needed to transmit the time stamp t_m after the *NOW*-message can be omitted if the master's time stamp is incorporated in the next *NOW*-message. Figure 6 shows the timing behaviour of the protocol over two rounds of synchronisation.

The proposed protocol provides failure resilience in case of omission as long as the amount of omitted packets is smaller than the estimated omission degree. Based on this assumption,

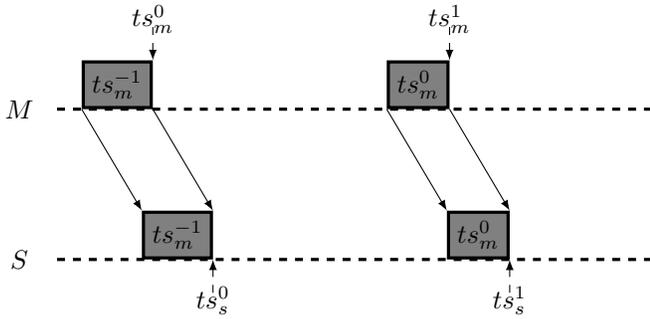


Figure 6. Timing diagram of 2 synchronisation rounds between a time master M and a time slave S .

the authors analytically derived (2) to compute the guaranteed precision π :

$$\pi \leq \frac{2\delta\Delta t(\rho + 2)}{(\Delta t)^2 - \delta^2}(\text{OD} + \text{INT}) + \delta \left(1 + \frac{(\Delta t)^2(1 + \rho) + \delta^2}{(\Delta t)^2 - \delta^2} \right) \quad (2)$$

In their experiments, they used a synchronisation interval $\Delta t = 10s$ on clocks with a drift of $\rho = 10^{-5}$. The maximum age of their second time stamp was $\text{INT} = 1s$. They measured a maximum jitter of the critical path of $\delta = 46\mu s$, resulting in a precision of $\pi = 150\mu s$, which fits with the results expected by (2) in case of an omission degree of $\text{OD} = 1$.

Even though multi-hop synchronisation is not covered in the paper, the approach of DMTS, see Section III-C, can also be used for CCS. Consequently, the multi-hop performance shall be similar to DMTS.

This approach enables continuous clock synchronisation without gaps in the time base suitable for real-time applications. Additionally, it provides better precision than the baseline 802.11 synchronisation mechanism without additional message overhead. It provides failure resilience and a guaranteed precision for a known omission degree. However, it is generally only useful for single-hop environments with a dedicated access point, since multi-hop performance will have the same problems as DMTS.

E. Probabilistic Clock Synchronisation Service (PCS)

The Probabilistic Clock Synchronisation Service by PalChaudhuri et al. [18] is an extension of RBS enabling a dynamic trade-off between synchronisation precision and message overhead. This approach transmits n NOW-messages in one synchronisation round, which are used to derive the skew of the sender's and the receiver's clock through linear regression. The results are combined and transmitted back to the receivers in range. By comparing their own data with the data received from the sender, they are able to adopt their own clocks. To derive the number of needed NOW-messages the authors assumed the synchronisation error to be normal distributed with zero-mean and a standard deviation of σ . Based on this distribution, the authors analytically derive the probability $P(|\epsilon| < \epsilon_{max})$ of the synchronisation error to be less than a specified value ϵ_{max} . For a specified probability of the synchronisation to be more precise than ϵ_{max} , the authors

derive the number n of message needed. This number heavily depends on the standard deviation σ of the normal distribution.

The multi-hop mechanism in this approach is based on a time transformation in a hierarchically structured network. If a node n_1 is synchronized to the master node n_0 broadcasting its time reference packets, this node can transform the received broadcasts based on arrival time and re-broadcast them to spread the synchronisation within the network. The error induced by the retransmission is again assumed to be normal distributed and analysed the same way as for the single-hop approach.

The robustness of the approach very much depends on the accuracy of the estimated distribution, if the real distribution of the synchronisation error in single-hop or multi-hop environments fits with the assumed distribution, the synchronisation will be very robust, since all failures as well as their results are already contained in the distribution. However, a mismatch of these will result in an unpredictable behaviour of the synchronisation. Consequently, the estimation of the probability distribution is the crucial part of this approach.

The approach provides a dynamic trade-off between message overhead and synchronisation precision even in multi-hop scenarios. Even though the trade-off depends on the standard deviation of the synchronisation error, the acquisition of this value was not covered by the authors. Additionally, only a mathematical proof without any simulation was conducted to evaluate the idea. Consequently, the authors never discussed the effects of non-normal distributed synchronisation errors.

F. Time Synchronisation in Ad-Hoc Networks (TSAN)

Römer's Time Synchronisation in Ad-Hoc Networks [19] is based on Christian's Algorithm [20]. It is a non-averaging internal synchronisation using pair-wise offset estimation. It estimates the round trip time of a message between sender and receiver, as visible through the transmission of e_0 in Figure 7 and ultimately tries to order events created by the system. Whereas Christian's Algorithm proposed a dedicated server for clients to communicate to, Römer attaches time stamps to events communicated in the network. Therefore, Römer's algorithm ideally induces a zero message overhead. However, not all events are acknowledged by the receiver, which might create large durations between events flowing in both directions between two nodes. This is mitigated by the insertion of additional dummy events in case the duration grows too large. This is shown through the communication between N_0 and N_2 in Figure 7.

TSAN supports multi-hop synchronisation directly, since it measures round-trip time from sender to receiver and back. The basic concept makes no assumption on the amount of hops between sender and receiver and works with an arbitrary amount of hops.

One of the major ideas of this approach is the usage of time transformations instead of clock synchronisation. In consequence, no node needs to set their clock to a certain value, but they only transform the timestamps of the events they receive to their own time domain using their estimated offset between sender and themselves. However, this offset estimation is limited to the events flowing by. As a result the first event flowing between two nodes cannot be transformed. The quality of the transformation depends on the amount of

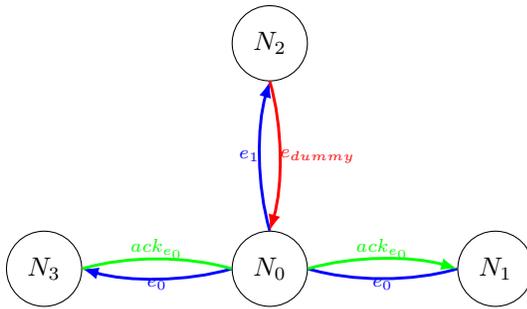


Figure 7. Schematic of a TSAN synchronisation. The blue events e_0 and e_1 are transmitted. e_0 is acknowledged back, showed as a green arrow. e_1 needs a dummy event (red) inserted, since no acknowledgement is done.

events. More events grant a better estimation of the offset between two nodes and therefore increase the transformation's precision.

Since TSAN assumes no bound on network delays of the events and the acknowledgements it uses, it cannot provide an upper bound of the synchronisations precision. This is mitigated by the usage of time intervals, which provide a duration instead of a singular time stamp. To achieve its main goal of ordering events, the time stamps are compared to infer the order of the events. However, it is possible that the intervals overlap and an ordering is not possible. In such cases, the system will provide a result of *MAYBE* for the order relation. The final order relation of TSAN for two time stamped events e_0 and e_1 containing the time stamps $e_0.ts = [t_0, t_1]$ and $e_1.ts = [t_2, t_3]$ is described in (3).

$$[t_0, t_1] < [t_2, t_3] = \begin{cases} YES & : t_1 < t_2 \\ NO & : t_3 < t_0 \\ MAYBE & : t_2 < t_1 \end{cases} \quad (3)$$

TSAN applies a very loose multi-hop synchronisation with an ideal message overhead of 0 in a multi-hop network. Unfortunately, the real message overhead is heavily dependent on the actual communication in the network and is therefore very hard to estimate for a real system. The used time intervals together with the *MAYBE*-results of the event ordering provide the protocol with a certain robustness against large synchronisation errors caused by message losses or unpredicted delays.

G. Summary

The individual problems and features of the protocols are summarized in Table I. As visible none of the described approaches fully solve the problem of multi-hop uncertainty aware clock synchronisation in wireless sensor networks. However, each approach contains individual features, which might enhance the performance of our approach.

- CS provided the idea of hierarchically structured synchronisation architecture using different synchronisation mechanism on the different layers of the hierarchy.
- DMTS introduced the usage of hardware access and knowledge on low-level behaviour to decrease the jitter of the critical path.
- CCS added the idea of a virtual clock following the clock of another node to provide a steady time

TABLE I. Comparison of the discussed time synchronisation protocols.

Protocol	Synchronisation Precision	Multi Hop Capability	Message Overhead	Robustness
CS	medium	inherent	$\mathcal{O}(n)$	medium
RBS	high	none	$\mathcal{O}(n^2)$	fragile
DMTS	high	possible	$\mathcal{O}(n)$	fragile
CCS	high	possible	$\mathcal{O}(1)$	robust
PCS	medium	inherent	dynamic	medium
TSAN	low	inherent	0 - $\mathcal{O}(1)$	medium

base without gaps. Additionally, it provided the differentiation of time stamp transmission and *NOW*-indication to shorten the critical path.

PCS proposed the estimation of the synchronisation error as a normally distributed random variable with a zero-mean and a known deviation. We will exploit that idea to estimate the uncertainty of our synchronisation.

TSAN provided the idea of time transformation between nodes on event reception, which we will exploit for our multi-hop synchronisation.

The proposed approach incorporates the beneficial properties of the different clock synchronisation mechanisms in a single clock synchronisation, which is uncertainty- and topology-aware and produces time intervals usable by an application. The next section describes it in detail.

IV. UNCERTAINTY AWARE CLOCK SYNCHRONISATION (UACS)

For an efficient synchronisation of clocks in WSN multiple parameters are important. On one hand, the synchronisation needs to be scalable, while on the other hand the overhead may not exceed a certain threshold to safe battery and prevent an overload of the network. Most of the approaches discussed in Section III favour one over the other. However, if we limit our self to certain base topologies better solutions might be found. One interesting topology is the cluster tree structure of IEEE 802.15.4 networks [21] in beacon-enabled mode. This mode divides the nodes in groups called Personal Area Networks (PANs), which have an individual coordinating instance managing the internal communication. The individual PANs communicate only through their respective coordinators, as visible in Figure 8. This hierarchical network structure may also be found in other types of networks like Bluetooth scatternets as proposed by the Bluetooth standard [22]. In the remaining section of the paper we consider an 802.15.4 network, with an already established cluster tree structure. The formation and the handling of dynamic changes in this structure are not considered in this paper.

Based on the initial assumption, that clock synchronisation may have a decreasing precision based on topological distance between nodes in the network, we propose a hybrid clock-synchronisation, consisting of a tight synchronisation mechanism for each individual PAN called Intra Cluster Synchronisation and a loose synchronisation mechanism between the individual PAN Coordinators, called Inter Cluster Synchronisation.

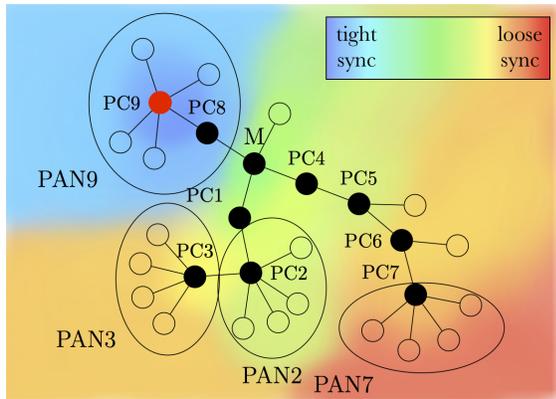


Figure 8. Example cluster tree structure of an IEEE 802.15.4 Network. The colours indicate the topological distance between each node and $PC9$.

A. Intra Cluster Synchronisation

The Intra Cluster Synchronisation is based on the CCS approach, see Section III-D. Therefore, each PAN slave $P_{s_j} \in \text{Slaves}$ has a virtual synchronized clock $VC_{s_j}(t)$. This clock uses the time stamps created by the node's hardware clock $C_{s_j}(t)$ and modifies it based on the current rate $r_{s_j,i}$:

$$VC_{s_j}(t) = r_{s_j,i} (C(t)_{s_j} - C(t_i)_{s_j}) + VC_{s_j}(t_i) \quad (4)$$

The task of the Intra Cluster Synchronisation is the estimation of the parameter $r_{s_j,i}$ for each slave at each synchronisation round i . The 802.15.4 standard allows a PAN Coordinator to attach additional information to the beacon frame. We use this to attach a 64bit time stamp $t_{c,i}$ to each beacon b_{i+1} transmitted by the coordinator. The attached time stamp represents the coordinator's time of successful transmission of the last beacon. This time stamp together with the local reception time of the last beacon $t_{s_j,i}$ is then evaluated by each slave P_{s_j} to compute a new rate $r_{s_j,i}$.

As described by DMTS, see Section III-C, hardware knowledge may be used to provide the needed local time stamps. The 802.15.4 standard provides the PD-Data.confirm primitive as a local event indicating completion of a transmission. The time of this event is used as the source of the time stamp $t_{c,i}$. On reception of the beacon each PAN slave P_{s_j} takes a local time stamp $t_{s_j,i+1}$. The network's tightness τ together with the internal computation time t_{comp} of the nodes limits the accuracy of the local time stamps. This computation time is mitigated by the PD-DATA.indication primitive of the 802.15.4 standard. Therefore, we consider t_{comp} to be very small in our approach and the time difference between creation of the local time stamps is bounded by τ .

After acquiring the time stamp for the actual synchronisation round the PAN slaves compute the offset $\delta_{j,i} = t_{c,i} - VC(t_{s_j,i})$ between their previous local time stamp $VC(t_{s_j,i})$ and the time stamp transmitted through the beacon $t_{c,i}$. This is used to compute a new rate $r_{s_j,i+1} = 1 + k_r \delta_{j,i}$ for the node's virtual clock to compensate the offset, with k_r being a proportional factor controlling the rate of adaption.

The Intra Cluster Synchronisation provides continuous clock synchronisation between the PAN Coordinator and its slaves. The overhead is minimal since no additional message

is necessary and the beacons are only slightly enlarged. The robustness of the synchronisation mainly depends on the used algorithm to detect a crash and reselect a PAN Coordinator.

B. Inter Cluster Synchronisation

Diverging from the Intra Cluster Synchronisation, see Section IV-A, a PAN Coordinator never modifies its own clock. Instead every event received by a PAN Coordinator P_{c_r} , which is transmitted by another adjacent PAN Coordinator P_{c_s} is transformed in the time domain of the PAN Coordinators clock $C_r(t)$, as proposed by TSAN, see Section III-F. To achieve this, the PAN Coordinator P_{c_r} needs to calculate a virtual clock $VC_{r,s}(t)$ for each adjacent PAN Coordinator P_{c_s} .

The virtual clocks are handled similarly to the Intra Cluster Synchronisation, since all beacons of all adjacent PAN Coordinators P_{c_s} are received by PAN Coordinator P_{c_r} . On reception of a beacon containing a time stamp $t_{s,i}$, P_{c_r} acquires a local time stamp $t_{r,s,i+1}$. This enables the computation of the offset $\delta_{r,s,i} = t_{s,i} - t_{r,s,i}$ between P_{c_s} and P_{c_r} . Afterwards, P_{c_r} updates the rate $r_{r,s,i} = 1 + k_r \delta_{r,s,i}$ for the virtual clock $VC_{r,s}(t)$ towards P_{c_s} . Therefore, each PAN Coordinator has an internal list of virtual clocks following the clocks of each adjacent PAN Coordinator as visible in Figure 9.

On reception of an event e_n from P_{c_s} containing a time stamp $e_n.ts_s$, P_{c_r} is able to transform the time stamp of the event to its own clock $C_r(t)$. The transformation is done by adding the offset between the Virtual Clock of the sender $VC_s(t)$ and the clock of the receiver $C_r(t)$ to the event's time stamp:

$$e_n.ts_r = e_n.ts_s + C_r(t) - VC_s(t) \quad (5)$$

In case of multi-hop communication the event's time stamp is always transformed to the receiver's clock domain before forwarding it to next hop. Consequently, all nodes only need to estimate the offset using a virtual clock for their directly adjacent neighbors. An example scenario is shown in Figure 9. In this picture three PAN Coordinators are in direct vicinity and estimate their offset using virtual clocks. An event is transmitted from PC_0 to PC_2 via PC_1 . During the forwarding of the event the time stamp of the event is adjusted by the estimated offsets to transform it to the local time domain of the current node.

C. Performance Estimation

The performance of the synchronisation depends on certain network and node parameters like the tightness of the network (including propagation speed and interrupt handling time) τ , the drift of the nodes ρ and the algorithm specific variable k_r . To analyse the behaviour of the algorithm we define two clocks: one for the transmitter of the beacon $C_s(t) = C_s(t_{s,i}) + (t - t_{s,i})(1 \pm \rho)$ and one for the receiver of the beacon $C_r(t) = C_r(t_{r,i}) + (t - t_{r,i})(1 \pm \rho)$. The offset between these two nodes is described as the difference between local clock of the sender and the Virtual Clock of the receiver as $o_{r,s}(t) = C_s(t) - VC_r(t)$. The receiver cannot observe this offset, but based on our algorithm it uses the visible offset $\delta_{r,s}(t) = o_{r,s}(t) \pm \tau$ based on the time stamp contained in the beacon. Consequently, the virtual clock's propagation formula may be rewritten to:

$$VC_r(t) = VC_r(t_{r,i}) + (C_r(t) - C_r(t_{r,i})(1 + k_r \delta_{r,s}(t_{r,i}))) \quad (6)$$

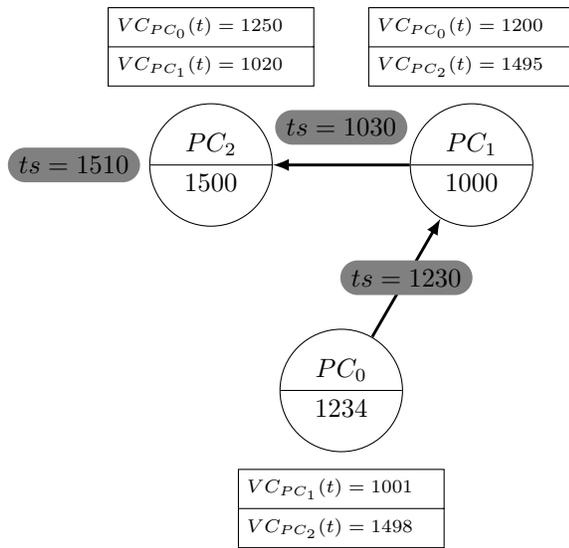


Figure 9. An example network composed of three PAN Coordinators with their respective virtual clocks estimating their offset towards each other. An event transmitted from PC_0 to PC_2 is shown in grey including its time stamp.

Entering $VC_r(t)$ in $o(t)$ enables replacing of $C_s(t) - VC_r(t_{r,i})$ with $o_{r,s}(t_{r,i}) + (t - t_{r,i})(1 \pm \rho)$. Together with the simplification of $C_r(t) - C_r(t_{r,i})$ to $(t - t_{r,i})(1 \pm \rho)$ one obtains:

$$o_{r,s}(t) = o_{r,s}(t_{r,i}) - (t - t_{r,i})(1 \pm \rho)k_r\delta_{r,s}(t_{r,i}) \pm (t - t_{r,i})2\rho \quad (7)$$

By substitution of $\delta_{r,s}(t_{r,i})$ and $(t - t_{r,i})$ to Δt as well as factoring out $o_{r,s}(t_{r,i})$ the following equation is achieved:

$$o_{r,s}(t) = o_{r,s}(t_{r,i})(1 - \Delta tk_r) \pm (\Delta t(k_r\rho o_{r,s}(t_{r,i}) + \rho(2 + \tau) + k_r\tau)) \quad (8)$$

To enable a normal flow of time, the rate of the Virtual Clock needs to be bigger than zero. Therefore, the value of k_r must not exceed $\frac{1}{\max(\delta_{r,s}(t))}$. The propagation of the offset may now be separated into an offset compensating part: $o_{r,s}^-(t) = o_{r,s}(t_{r,i})(1 - \Delta tk_r)$ and an offset increasing part: $o_{r,s}^+(t) = \pm \Delta t(k_r\rho o_{r,s}(t_{r,i}) + \rho(2 + \tau) + k_r\tau)$. The offset decreasing part will converge towards zero, because it resembles the geometric sequence $a^{i+1} = a^i q, 0 < q < 1$. The offset inducing part can be separated into an offset dependent and offset independent part. The value of the offset dependent part $\Delta tk_r\rho o_{r,s}(t_{r,i})$ depends very much on the offset between sender and receiver at the last synchronisation and the time passed since this synchronisation. For tightly synchronized nodes the impact of this part will tend towards zero.

The guaranteed precision of the method for tightly synchronized nodes is the absolute value of the offset independent part of $o_{r,s}(t)$. The resulting precision of the Intra Cluster Synchronisation is $\pi_{intra}(t) = \Delta t(\rho(2 + \tau) + k_r\tau)$.

The offset independent part is $\Delta t(\rho(2 + \tau) + k_r\tau)$. In case no beacon loss occurs Δt will be approximately two times the beacon Δt_b interval, because every beacon contains

the sender's time stamp of the last round and the maximum offset is always reached directly before the next beacon arrives. The resulting precision in case of OD beacon omissions will therefore be:

$$\pi_{intra} \leq (2 + OD)\Delta t_b(\rho(2 + \tau) + k_r\tau) \quad (9)$$

The value k_r can be viewed as a trade-off factor choosing between fast synchronisation and robust synchronisation. This is caused by the presence of the factor in the offset compensating part, where it decreases the existing offset stronger if it is bigger. On the other hand, bigger k_r values will increase the results of the offset increasing part depending on the values of ρ and τ .

For the offset estimation of the adjacent neighbours the Inter Cluster synchronisation uses the same approach as the Intra Cluster Synchronisation uses. Therefore, the multi-hop synchronisation precision π_{inter} for tightly synchronized nodes can be bounded using the hop count h :

$$\pi_{inter} \leq h(2 + OD)\Delta t_b(\rho(2 + \tau) + k_r\tau) \quad (10)$$

For loosely synchronized nodes the offset of the last synchronisation will be a relevant issue. However, this is only a problem in mobile systems, since static systems will always compensate the offset as long as the time between synchronisations is not bigger than the maximum considered offset. A direct consequence is that the synchronisation precision in multi-hop scenarios can be greatly enhanced if routes of tightly synchronized nodes are chosen.

D. Mobility

Mobility influences the networks topology and therefore, the association of slaves to masters and the interconnectivity between masters. Consequently, if a slave loses connection to its respective master it is entering the orphaned state and may only communicate again after it re-associated. If the node re-associates with the same master the node's maximum offset $o_{r,s}(t)$ will depend on the time between loss of link and the reception of the next beacon Δt and the offset of master and slave at the last synchronisation as described by (8). A tightly synchronized slave will increase its offset based on the time of the last synchronisation and the drift of the nodes. A loosely synchronized node will additionally increase the offset by a value proportional to last offset ($o_{r,s}(t_{r,i})$), time since last synchronisation (Δt), drift (ρ) and network tightness (τ). Depending on the value of k_r , fast moving slaves will never reach a tightly synchronized state, because they are switching masters faster than the algorithm can compensate the initial offset. On the other hand, a high k_r will enforce the induced errors when in orphaned state and the slave reconnects to the same master.

In contrast to the independence of the movement of slaves, the movement of masters have an impact on the whole PAN created by this master. Therefore, the master election mechanism should select slowly moving nodes as masters, to increase the time of the algorithm to converge the offset towards zero.

The Inter Cluster Synchronisation handles mobility well, since the mobility of a node in the local neighbourhood of P_{c1} does not change P_{c1} 's virtual clocks of the adjacent

nodes. Therefore, the transformation of the events is independent of each other. Since there is no hierarchy between the PAN Coordinators, the movement of each node only effects the offset estimation towards its neighbours and the estimation of the neighbours towards this node. New nodes in a neighbourhood start with an infinite uncertainty in the offset estimation since they have not yet established an offset estimation towards their neighbours. While the nodes receive beacons from adjacent masters, they improve the offset estimation and decrease the uncertainty. Consequently, we explicitly specify the offset estimation uncertainty in a time interval $ti = [ts \pm \alpha], ts \in R^+, \alpha \in R^+$ replacing the time stamp $ts \in R^+$. Furthermore, each hop modifies the interval bounds by the currently estimated uncertainty of the synchronisation $\alpha_{r,s}$, as described by (11):

$$e_n \cdot \alpha_r = e_n \cdot \alpha_s + \alpha_{r,s} \quad (11)$$

E. Estimating the Uncertainty

The estimation of the current uncertainty of the synchronisation of the virtual clocks is difficult. Multiple factors influence the actual uncertainty in the synchronisation, like beacon losses and the current drift of the individual clocks. In our approach the synchronisation error $\epsilon_{r,s,i}$ of synchronisation round i between two adjacent PAN Coordinators P_{c_s} and P_{c_r} is characterized by their offset $\delta_{r,s,i+1}$ at beginning of synchronisation round $i + 1$.

Following PCS, see Section III-E, we model the synchronisation error to be a zero-mean Gaussian distribution $N(0, \sigma)_{r,s}$. To estimate the standard deviation we use the synchronisation errors of the previous n synchronisation rounds as sample set $E_{r,s} = \{\epsilon_{i-n}, \epsilon_{i-n+1} \dots \epsilon_i\}$. We estimate the standard deviation $\sigma_{r,s}$ of our zero-mean Gaussian based on the sample set. Based on this we compute the confidence interval $\left[\bar{x} \pm z\left(\frac{1+\gamma}{2}\right) \frac{\sigma}{\sqrt{n}} \right]$ of the synchronisation with typical probability γ . The value $z\left(\frac{1+\gamma}{2}\right)$ represents the $\frac{1+\gamma}{2}$ -quantile of the standardised normal distribution. The resulting size of the confidence interval $\alpha_{r,s} = z\left(\frac{1+\gamma}{2}\right) \frac{\sigma}{\sqrt{n}}$ represents our current uncertainty estimation, which is added to current uncertainty of the event's time interval.

The complexity of this computation is only dependent on n , which represents a trade-off between estimation accuracy and memory and computation overhead. The quantile of the standardised normal distribution is a pre-defined constant, characterising the accuracy of the estimation.

F. Compatibility between Time Intervals and Time Stamps

As described in the introduction (Section I), WSN aim not only to distribute the acquired sensor data, but also need to process the data in form of events. To this end, Liebig et al. [23] described a way to combine multiple events even though their time stamps might not be exact. To achieve this they extended a time stamp to a time interval and derived an order relation $<$ for time intervals described in (12). Together with a known uncertainty of the event's time stamp, ordering might be possible even in loosely synchronized systems. However, this transition induced a partial order through the order relation. Consequently, there might be situations in which two events cannot be ordered. This is the case if the intervals of the events'

time stamps overlap. The resulting partial order relation is similar to the one used by TSAN.

$$[ts_0 \pm \alpha_0] < [ts_1 \pm \alpha_1] \Leftrightarrow ts_0 + \alpha_0 < ts_1 - \alpha_1 \quad (12)$$

Compared to TSAN III-F, our aim is to provide all operations on time intervals that are available for time stamps. This enables applications to handle time intervals in the same manner as classical time stamps, but with the awareness of the induced uncertainty. Our time-interval algebra ($[ts \pm \alpha], \{+, -, \times, \cdot, ()^{-1}, <\}$) is based on interval arithmetic [24] and the proposed partial order relation. The difference to the general interval arithmetic is the definition of the inverse operation ($()^{-1}$) and the ability to scale the time interval by a constant factor (\cdot). Both operations are very useful to combine events. One example of such a composition is the deduction of events containing the speed (e_s) of an object based on events containing the position (e_p^i) of the object, as described by Steup et. al [25]. The computation necessary for such a composition is shown in (13).

$$e_s \cdot speed = (e_p^1 \cdot pos - e_p^0 \cdot pos) (e_p^1 \cdot ts - e_p^0 \cdot ts)^{-1} \quad (13)$$

This type of computation is typical for cyber-physical systems containing physical processes. In general, these processes may be described by differential equations. As an approximation, we enable the computation of difference quotients if the basic events can be ordered. As defined by the partial order, two events are ordered whenever their time intervals are disjoint. Consequently, the time interval created by the subtraction of their time stamps may never contain zero. As a result we simplified the inverse operation compared to general interval arithmetic to (14).

$$[ts \pm \alpha]^{-1} = [ts^{-1} \pm \alpha^{-1}] \quad (14)$$

$$ts^{-1} = \frac{1/2}{ts + \alpha} + \frac{1/2}{ts - \alpha} \quad (15)$$

$$\alpha^{-1} = \frac{1/2}{ts - \alpha} - \frac{1/2}{ts + \alpha} \quad (16)$$

The resulting time interval algebra establishes a partially ordered vector space, which is easy to compute even for deeply embedded systems. The additional operations enable the computation of differential equations, which are necessary to describe most physical processes. The transformation back to classic time stamps is easily possible by omitting the uncertainty part of the interval.

V. EVALUATION

The uncertainty aware hybrid clock synchronisation system was evaluated with a simulation in the Omnet++ Network Simulator [26] version 4.2 and a small scale WSN composed of six nodes.

A. Simulation Setup

For the simulated evaluation we used the INETMANET network model [27] as well as the MiXiM model [28]. The implementation is distributed over two layers of the ISO/OSI stack. One part is located at layer 5 of the ISO/OSI stack and handles the transformation of time stamps for the Inter Cluster synchronisation. The other is situated at layer 2 to gather high precision time stamps. Both layers are connected through a cross layer communication.

Our evaluation focuses on the Inter Cluster Synchronisation, since our simulation experiments shall investigate the scalability of the approach. The single-hop performance will be evaluated by the real world experiments. We evaluate two main aspects of the Inter Cluster Synchronisation. The first considers the influence of the beacon period on the precision of the synchronisation. This test will provide information on the trade-off between message overhead and synchronisation quality. The second test investigates the influence of the communication topology on the reachable multi-hop precision. It will evaluate the usability of the provided time stamps for smaller and longer routes. All tests used the internal 64 bit `simtime` of Omnet++ as reference for the synchronized clocks to evaluate the synchronisation error. The `simtime` was modified by a randomly initialized drift $\rho \leq 10^{-5}$, to provide a realistic clock for each node. The test considered 1000 randomly created routes between nodes in the network, which were created by an optimal routing algorithm.

Our simulation environment considers beacon losses, created by the collision of transmitted beacons of adjacent coordinators, and the resulting lack of information for the time synchronisation. However, we did not transmit data events in the simulation. This decouples our simulations from the used MAC Algorithm and its parameters. Consequently, the simulations consider an optimal MAC-Algorithm preventing all collision between beacons and events in the network.

Additionally, we did not simulate the interrupt handling time possibly decreasing the tightness of the network, since we estimated this time to be smaller than $1\mu s$ because of our optimization described in Section V-D.

B. Beacon Interval Analysis

The beacon interval analysis considered a rectangular grid of 50 PAN Coordinators. The area in which the nodes were distributed was $5000m$ times $5000m$. We used the 2.4GHz specification of the 802.15.4 standard at channel 11 with a maximum transmission power of $1mW$. The thermal noise was fixed at $110dBm$ and the receiver's sensitivity was set to $-85dBm$. Our simulation sweep started with a BO parameter of 8 up till the maximum allowed value of 14. The resulting beacon interval can be computed by $BI = \frac{16 \cdot 60S \cdot 2^{BO}}{SymbolRate}$. The `SymbolRate` of the 2.4GHz band of $65.2 \cdot 10^3 \frac{S}{s}$ results in beacon intervals from $3.8s$ to $241.2s$.

Figure 10 shows a Box-Whisker plot of the simulation's results. The boxes represent the bounds, where 50% of all values are included. The lines represent the interval containing 75% of all values and remaining data points are included as points. As visible with linear increasing BO values the mean synchronisation error increases exponentially. This is to be expected because the beacon interval also increases exponentially. Additionally, one observes a large standard

deviation independent of the hop count. This is caused by the unsynchronized beacons of the individual PAN Coordinators, which might collide and therefore increase the real beacon interval. Furthermore, the data base is better for smaller hop counts, since in the given scenario short routes are much more probable than longer routes.

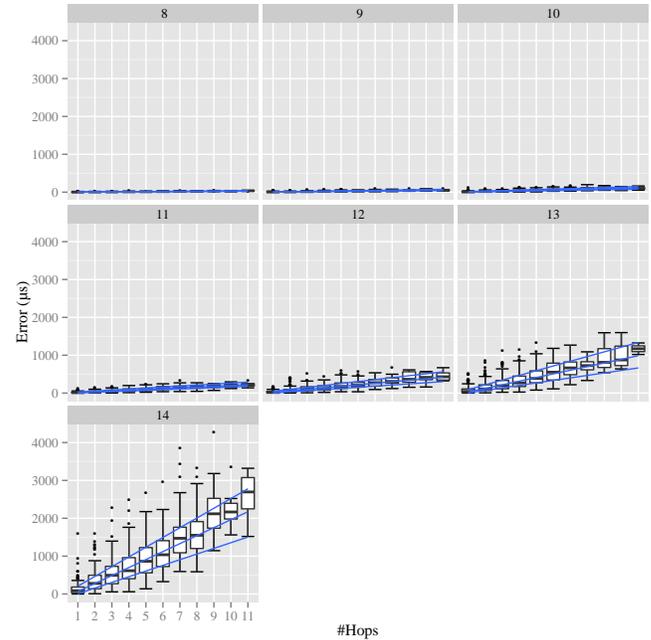


Figure 10. Box-Whisker plot of the precision of a 50 node grid network with varying BO values.

This test proved the expected direct correlation between the beacon interval and the synchronisation precision. Therefore, this value is to be considered critical for the performance of the system. Following our analyses of Section IV-C, the worst case performance should be better than $h2(0.0147 \cdot 2^{BO} \cdot 2.1 \cdot 10^{-11}s)$. In all our experiments we never crossed this analytical worst case bound. In the case of 1-hop long routes with a beacon order of 14 the estimated worst case precision is $9.6ms$, where our experiment showed a worst case result of $1.6ms$. For beacon order 13 and a hop count of three we achieved a worst case precision in our experiment of $1.2ms$ and analytically derived a worst case precision of $14.4ms$. We believe these large differences are created by the random drift, which in most cases will not create the largest possible offset.

C. Topology Analysis

Our second evaluation considers the performance of the system in different topologies. This is interesting, because topologies might have an influence on the length of the routes, as well as the collision probability of the beacon frames. Therefore, we consider four basic topologies with 200 nodes each. We choose a linear (c.f. Figure 11a), a circular (c.f. Figure 11b), a grid (c.f. Figure 11c) and a randomly generated (c.f. Figure 11d) topology. For this scenario we use the same parameters as for the Beacon Interval Analysis, see Section V-B, but with a static BO parameter of 8.

As visible in Figure 12, the linear topology showed a maximum synchronisation error of $875\mu s$. This is mainly

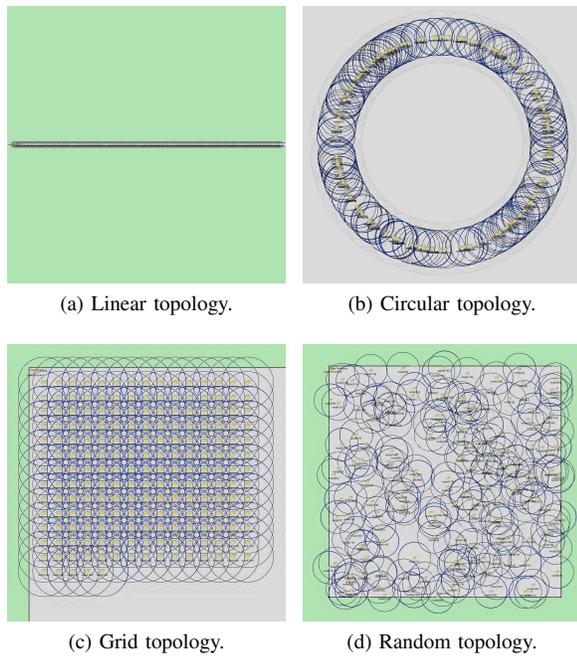


Figure 11. Different evaluated topologies with 200 nodes each.

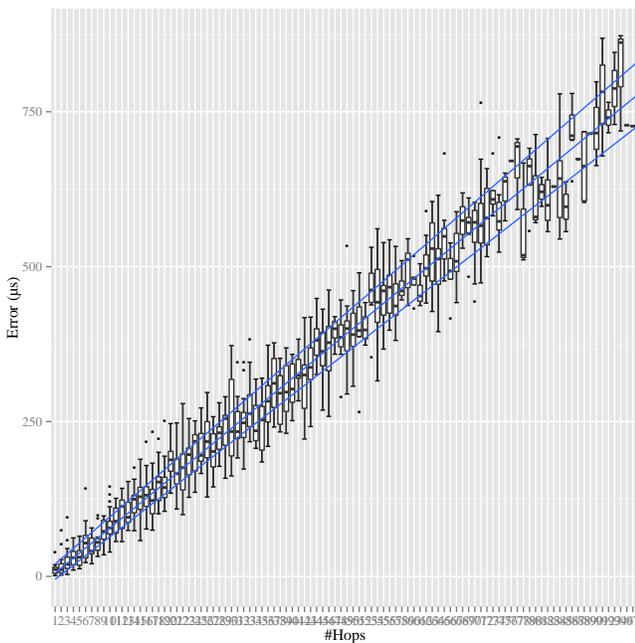


Figure 12. Box-Wisker plot of the synchronisation precision of a 200 Node linear topology.

caused by the extremely long routes created by the evaluation. The maximum hop count was 69 and for each pair of randomly selected nodes there is only one possible route. Consequently, a badly synchronized node will have a large influence on the resulting precision. Considering our worst case analyses we still performed far better than the worst case of 103.8ms. This is caused by the very small possibility of randomly finding

a long route containing 69 nodes with a maximum drift or a large amount of lost beacons.

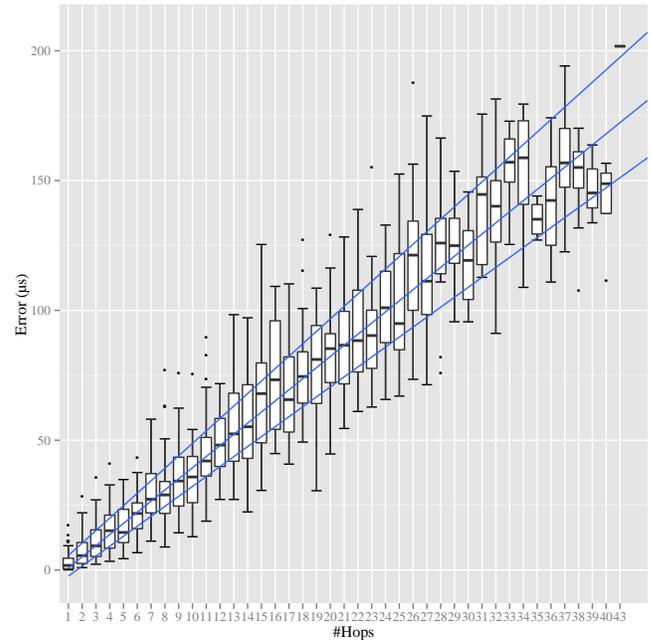


Figure 13. Box-Wisker plot of the synchronisation precision of a 200 Node circle topology.

The circular topology was expected to show similar results like the linear topology, but it performed a lot better, as visible in Figure 13. The maximum hop count was 40 with a typical synchronisation error of 150µs. Compared with the linear topology, which provided a synchronisation error of approximately 500µs, it outperformed the linear topology. This is caused by the larger amount of routes that were possible to connect two randomly selected nodes. A badly synchronized node does not have so much influence anymore, since it is not as likely a part of the random route. Additionally, collisions of beacons are quite unlikely, because only a small number of nodes are in the vicinity of each other. In this experiment one route of 14 hops had a comparably large synchronisation error of 175µs which was still smaller than the worst case approximation of 2107µs.

The grid topologies (Figure 14) showed slightly worse performance compared to the circle topology. This is caused by the larger probability of beacon collisions caused by a higher density of nodes. At the same time the maximum hop count is only 25, which created a maximum synchronisation error of approximately 150µs. The circle topology showed only a synchronisation error of approximately 100µs for routes of the same length.

In Figure 15, the performance of the random topology is visible. It shows a large deviation of individual results, which is caused by individual collisions of beacons. This problem is very dependent on the local setup of nodes around a PAN. Therefore, it is very hard to estimate and may only be observed on runtime by an uncertainty evaluation. Such hot spots are also quite likely to be contained in a randomly generated route, since hot spots contain more nodes compared to the

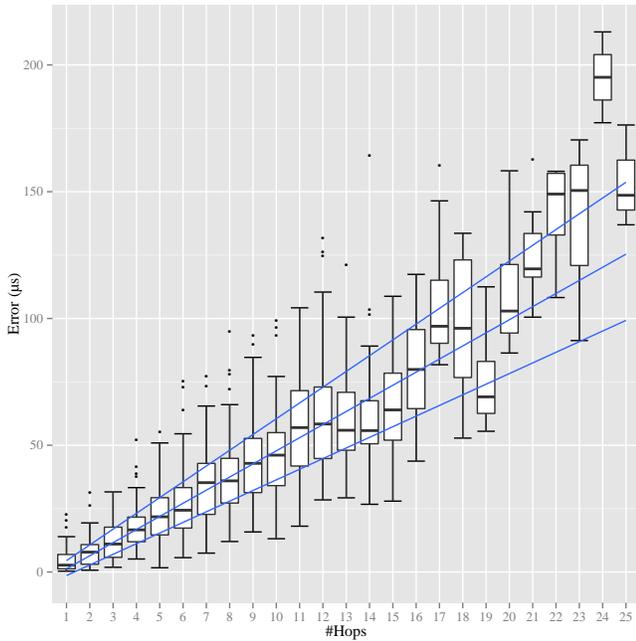


Figure 14. Box-Wisker plot of the synchronisation precision of a 200 Node grid topology.

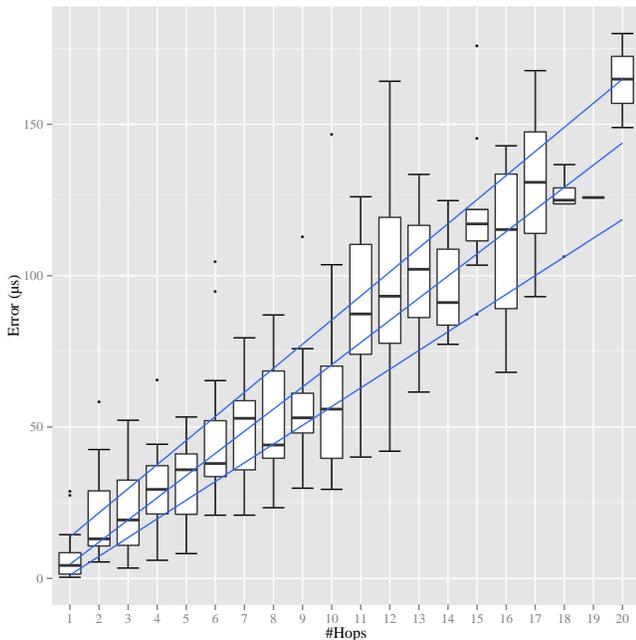


Figure 15. Box-Wisker plot of the synchronisation precision of a 200 Node random topology.

surrounding areas. These facts decrease the achievable mean precision. In this experiment a route of length 12 showed a large error of $161\mu s$ compared to the main quantile of the 12 hop long routes. The worst case analyses predicted a maximum error of $1806\mu s$.

TABLE II. Synchronisation error of the simulation of different topologies in μs .

#Hop Count	Topology	Mean	Standard Deviation
1	Random	7.069008	8.495
1	Grid	5.067219	5.454
1	Linear	12.181786	9.492
1	Circle	3.327730	3.993
6	Random	47.401797	23.945
6	Grid	26.574773	13.489
6	Linear	55.861373	31.209
6	Circle	21.579193	8.327
11	Random	89.191069	27.941
11	Grid	57.463772	19.879
11	Linear	91.408097	23.836
11	Circle	45.693382	15.802
16	Random	110.255113	29.990
16	Grid	80.882388	20.756
16	Linear	131.582874	31.220
16	Circle	73.628941	21.878

Table II shows an overview of the results of our evaluation of the different topologies. This table shows that the performance of the individual topologies towards the mean for each hop is $\pm 50\%$. However, the results for the standard deviation of the tests for equally long routes show a larger difference between the individual topologies. Additionally, the value of the standard deviation is for all single-hop routes approximately the same as the mean error. This clearly indicates a need for a runtime uncertainty estimation, as described in Section IV-E. All topologies supported our mathematical analyses of the worst case synchronisation error since no experiment exceeded the worst case.

As expected, in all simulations the linear topologies have the largest mean error, which is caused by the highest collision probability of the beacons. The random topology performed the second worst, which is caused by local hot spots in the topology with a lot of nodes increasing the probability of beacon losses. All topologies clearly showed a linear relation between the mean error and the hop count. This is to be expected, since the synchronisation error in the vicinity of each PAN is statistically the same within each topology. The summation of the uncertainties matches very well with the increasing error in the simulation. The beacon loss probability is network specific and does not only depend on the topology, but also on the density of nodes. This probability together with the mean length of routes in the network is the major influence towards the synchronisation precision.

From these experiments, we concluded that the basic assumptions were valid. Additionally, we observed that regular non-linear topologies provide better synchronisation results. As a next step, we want to evaluate the performance of the approach on real hardware.

D. Small Scale Wireless Sensor Network Setup

To evaluate the correctness of our assumptions in the simulation we choose a small wireless network of 6 nodes to compare the results with the simulation and related work. Our test network is composed of three PAN Coordinators, two Slaves and a Raspberry Pi Model A [29]. The nodes are

Cortex-M3 based devices from dresden elektronik [30] using a 2.4 GHz 802.14.5 transceiver of Atmel [31]. They run with an internal crystal oscillator and a PLL providing a clock speed of 32 MHz, which is divided by 32 to provide an internal clock with a granularity $g = 1\mu s$. The used 18.432MHz oscillator has a drift of $\rho = 3 \cdot 10^{-5}$. We implemented our approach using hardware timers of the Cortex-M3 microcontroller taking a time stamp on each interrupt in hardware. Using this approach the interrupt delay between reception of the packet and the generation of the time stamp should be decreased below $1\mu s$. The propagation time of the wireless signal in the typical range of the AT86RF232 is $t \leq 34ns$. The time between the reception of a packet and the generation of the interrupt by the transceiver is given as $t_{irq} = 9\mu s$. Because the interrupt latency of the transceiver is the same for sender and receiver and, according to data sheet, is constant, it may be omitted. The resulting tightness of our beacon network can be assumed to be $\tau \approx 1\mu s$.

The Raspberry Pi uses a Preempt-RT patched Linux kernel [32] with a real-time enabled listening program. The coordinators and the slaves use the Atmel MAC stack [33] to handle time stamp generation, association and beacon transmission.

Both slaves and the three coordinators are connected to the Raspberry Pi through a GPIO cable as visible in Figure 16. Additionally, each device is connected to an evaluation PC to log the time stamps. Each coordinator establishes its own PAN, but also receives the neighbouring PAN's beacons.

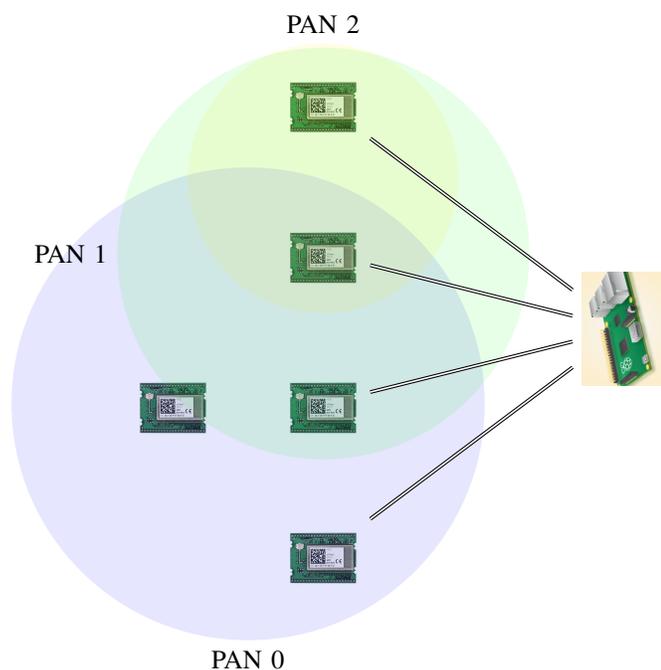


Figure 16. Small scale Wireless Sensor Network composed of dresden elektronik nodes [30] and a Raspberry Pi [29] used to evaluate the hybrid synchronisation.

Whenever a PAN Coordinator transmits its beacon it also logs its internal time to the evaluation PC and toggles its GPIO. On reception of a beacon each node transmits its virtual clock

time stamp following the sender to the evaluation PC and toggle their GPIO-Pin. The Raspberry Pi monitors continuously the GPIO-Pins and takes a time stamp on each change. The resulting pair of Raspberry Pi and Cortex-M3 time stamps are correlated and analysed to provide an accurate offset estimation of the virtual clocks against the internal clocks of the nodes. Since beacons are transmitted unsynchronized on the different coordinators, we use linear interpolation to compute time stamps in between measured values.

The benefit of this setup is the minimal critical path, which is established by the GPIO connection. The toggling of the pin on the device is instantaneous. The available low-level access library of the Raspberry Pi provides extremely small latencies accessing the pins. Together with the used real-time program the measurement error should be smaller than $\approx 1\mu s$.

E. Single-Hop Synchronisation

This setup evaluated the single-hop synchronisation mechanism. It is used as a baseline to verify the correctness of the parameters of the simulation. Therefore, the results should be close to the one hop results of the simulation. All coordinators were running and transmitting beacons, but only the two slaves ran the virtual clock towards their PAN Coordinator. The experiment was run for 5 minutes with a beacon interval of 7.5s and 15s representing a BO value of 9 and 10, respectively.

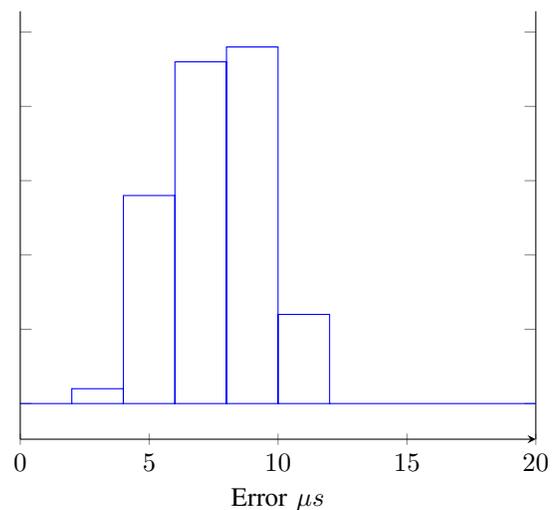


Figure 17. Histogram of the achieved precision of the internal synchronisation with a beacon interval of 7.5s.

The mean precision of the internal synchronisation with a beacon interval of 7.5s, as visible in Figure 17, was approximately $8\mu s$, which fits very good to the simulated results. The deviation was approximately $\pm 1\mu s$ with the maximum error being $13\mu s$. This was less than the deviation measured in the simulation. An explanation is the smaller network size creating less beacon loss. The distribution of the values is approximately normal distributed, fitting to our assumption used in the uncertainty estimation.

Figure 18 shows the results of the experiment with a beacon interval of 15s. As visible the mean precision was also $8\mu s$ like in the previous experiment. The deviation of the precision is bigger being approximately $2\mu s$ with the maximum being

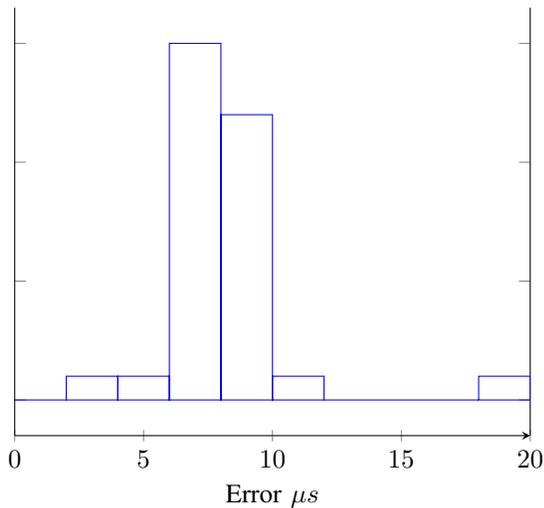


Figure 18. Plot of the achieved results of the single-hop synchronisation with a beacon interval of 15s.

$20\mu s$. This is again better than simulation results, possibly because of less beacon loss and the crystal oscillator being better than described in the data sheet.

F. Multi-Hop Synchronisation

The evaluation of our Inter Cluster Synchronisation used only the three coordinators. All coordinators were broadcasting beacons in this setup and ran a virtual clock for each neighbouring node. We evaluate the virtual clock values of the different nodes and the time stamps of the internal clocks of the nodes at periodic intervals. The evaluation interval is the same as the beacon interval. No real event was routed through the network, since the used MAC stack provides no means of multi-hop communication. Therefore, we simply added the estimated offsets of the virtual clocks of the two pairs of nodes and compared them to the offsets of the internal clocks of the first and the last coordinator. This scenario used a fixed beacon interval of 7.5s.

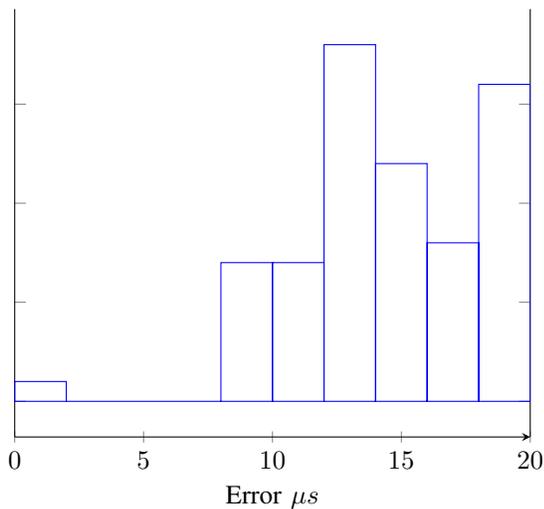


Figure 19. Plot of the achieved results of the inter cluster synchronisation with a beacon interval of 7.5s.

As visible the mean precision was $16\mu s$ with a large deviation of $\pm 5\mu s$ and a maximum offset of $20\mu s$. This result fits very well with the expected simulated and expected results. The error is approximately doubled compared to the single-hop scenario. Additionally, the deviation has increased by the same margin.

The results of our small scale wireless sensor network experiments fully support our simulated and expected performance of our approach. The baseline performance fitted very well with the exception of the larger beacon interval, which strangely showed nearly the same results.

G. Comparison with related protocols

Since the environments of the different described protocols differ, we compare our approach to protocols, with available multi-hop synchronisation data. DMTS provided a mean synchronisation error of $32\mu s$ for one hop and $46\mu s$ for two hop communication. Our approach performed better for single-hop ($8\mu s$) and two-hop synchronisation ($16\mu s$). In comparison to DMTS we do not need a possibly incorrect model of the latency induced by the interrupt handling of the nodes. Additionally, the granularity of the internal clocks of our nodes was far better ($g = 1\mu s$) than the ones used in the experiments evaluating DMTS ($g = 32\mu s$). Therefore, the results are not directly comparable. TSAN showed a mean synchronisation error of $200\mu s$ for one hop and $1113\mu s$ for six hop communication. Our approach performed better in both cases (on average $8\mu s$ and $25 - 37\mu s$ worst case $27\mu s$ and $110\mu s$). However, Römer et al. considered an unstructured abstract network, whereas we exploited the structure and the hardware of the network to increase the synchronisation precision without message overhead. Especially, the periodicity of the beacons enabled continuous synchronisation, which was not available to Römer's system. Mock et al. showed a single-hop synchronisation of approximately $150\mu s$, which was mainly caused by the driver abstraction and the interrupt handling of the used operating system, since they considered an experimentally derived tightness of the network $\tau = 46\mu s$. Our performance stems from the bare-metal implementation and the good local clock increasing the networks tightness to $\tau \approx 1\mu s$.

VI. CONCLUSION

This paper presents and evaluates a novel hybrid clock synchronisation approach that provides tight synchronisation for local clusters of nodes as well as looser synchronisation in multi-hop scenarios. The message overhead is minimal since existing periodic beacon messages of the 802.15.4 beacon-enabled mode are used to transmit the synchronisation data. To handle the different synchronisation precisions, uncertainty awareness is added to enable applications to decide in the case of ambiguity.

The evaluation is done using the well-established network simulator Omnet++ and a small scale wireless sensor network verifying the results and parameters of the simulation. The results of both experiments match very well with the theoretical concepts and the expected performance of the system. The system provides a baseline performance of $8\mu s$ with a deviation of $\pm 2\mu s$ in a single-hop environment in simulation and real test. In multi-hop scenarios a linearly decreasing precision can be observed depending on the hop count. The results show

the large difference between worst case approximation and real performance together with the large deviation between individual synchronisation results. This clearly indicates the need for uncertainty awareness in the delivered time stamps.

In future work, we want to evaluate the clock synchronisation in a real scenario with more realistic wireless sensor networks to evaluate the influence of unforeseen interference especially in non-regular topologies. Additionally, we want to investigate the effect of different MAC-Algorithms on the synchronisation quality.

ACKNOWLEDGEMENTS

We thank our students Anita Hrubos and David Bodnar for their effort in implementing, testing and evaluating the protocol on our microcontrollers. Additionally, we want to express our gratitude towards Andy Breuhan, who implemented and evaluated the approach in the OMNET++ network simulator.

REFERENCES

- [1] C. Steup, S. Zug, J. Kaiser, and A. Breuhan, "Uncertainty Aware Hybrid Clock Synchronisation in Wireless Sensor Networks," in The 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), IARIA. Rome, Italy: Curran Associates Inc., Aug. 2014, pp. 246–251.
- [2] Y. Abe, "Safecast or the Production of Collective Intelligence on Radiation Risks after 3.11," in The Asia-Pacific Journal, vol. 12, Issue 7, No. 5, Feb. 2014, pp. 1–6.
- [3] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, Sep. 2005, pp. 1214–1217.
- [4] F. Michahelles, P. Matter, A. Schmidt, and B. Schiele, "Applying wearable sensors to avalanche rescue," Computers & Graphics, vol. 27, no. 6, Dec. 2003, pp. 839–847.
- [5] K. Römer and F. Mattern, "The design space of wireless sensor networks," IEEE Wireless Communications, vol. 11, no. 6, Dec. 2004, pp. 54–61.
- [6] P. Verissimo, Paulo and L. Rodrigues, Distributed Systems for System Architects, 1st ed. Kluwer Academic Publishers, 2001, ch. 2.
- [7] P. Hetzel, "Time dissemination via the LF transmitter DCF77 using a pseudo-random phase-shift keying of the carrier," in Proceedings of the 2nd European Frequency and Time Forum (EFTF), 1988, pp. 351–364.
- [8] P. Dana, "Global Positioning System (GPS) Time Dissemination for Real-Time Applications," Real-Time Systems, vol. 12, no. 1, 1997, pp. 9–40.
- [9] H. Kopetz, Real-Time Systems: Design Principles for Distributed Embedded Applications, 2nd ed. Springer Science+Business Media, 2011, ch. 3.
- [10] J. Burbank, D. Mills, and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification," Internet Engineering Task Force (IETF), Tech. Rep. RFC 5905, Jun. 2010.
- [11] "IEEE Standard for a Precision Clock Synchronisation Protocol for Networked Measurement and Control Systems," IEEE, Tech. Rep. Standard 1588-2002, 2002.
- [12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronisation using reference broadcasts," in SIGOPS, vol. 36, no. SI. New York, NY, USA: ACM, Dec. 2002, pp. 147–163.
- [13] P. Verissimo, L. Rodrigues, and A. Casimiro, "Cesiumspray: a precise and accurate global time service for large-scale systems," Real-Time Systems, vol. 12, no. 3, 1997, pp. 243–294.
- [14] S. Ping, "Delay measurement time synchronisation for wireless sensor networks," Intel Research Berkeley Lab, Tech. Rep. IRB-TR-03-013, 2003.
- [15] M. Mock, R. Frings, E. Nett, and S. Trikaliotis, "Continuous clock synchronisation in wireless real-time applications," in Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems, Oct. 2000, pp. 125–132.
- [16] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE, Tech. Rep. IEEE Standard 802.11, 1997.
- [17] M. Gergeleit and H. Streich, "Implementing a distributed high-resolution real-time clock using the CAN-bus," in Proceedings of the 1st international CAN-Conference. Mainz, Germany: Can-in-Automation (CIA), 1994.
- [18] S. PalChaudhuri, A. Saha, and D. B. Johnson, "Probabilistic clock synchronisation service in sensor networks," in IEEE Transactions on Networking, vol. 2, no. 2, 2003, pp. 177–189.
- [19] K. Römer, "Time synchronisation in ad hoc networks," in Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, ser. MobiHoc '01. New York, NY, USA: ACM, 2001, pp. 173–182.
- [20] F. Cristian, "Probabilistic clock synchronisation," in Distributed Computing, vol. 3, no. 3. Springer, Sep. 1989, pp. 146–158.
- [21] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," Tech. Rep. IEEE Standard 802.15.4, 2006.
- [22] Bluetooth SIG, "Specification of the Bluetooth System," Tech. Rep. 4.0, June 2010.
- [23] C. Liebig, M. Cilia, and A. Buchmann, "Event Composition in Time-Dependent Distributed Systems," in Proceedings of the Fourth IECIS International Conference on Cooperative Information Systems, ser. COOPIS '99. Washington, DC, USA: IEEE Computer Society, Sep. 1999, pp. 70–78.
- [24] R. E. Moore, R. B. Kaerfott, and M. J. Cloud, "The Interval Number System," in Introduction to Interval Analysis, 1st ed. Society for Industrial and Applied Mathematics, Jan. 2009, pp. 7–18.
- [25] C. Steup, S. Zug, and J. Kaiser, "Achieving Cooperative Sensing in Automotive Scenarios through Complex Event Processing," in The 7th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM), IARIA. Porto, Portugal: Curran Associates Inc., Sep. 2013, pp. 26–30.
- [26] G. Pongor, "OMNeT: objective modular network testbed," in Proceedings of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems. San Diego, CA, USA: Society for Computer Simulation International, Oct. 1993, p. 323–326.
- [27] A. Ariza and A. Triviño, Simulation of Multihop Wireless Networks in OMNeT++. IGI Global, 2012, pp. 140–158.
- [28] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating Wireless and Mobile Networks in OMNeT++ the MiXiM Vision," in Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops. Brussels, Belgium: Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Mar. 2008, p. 71:1–71:8.
- [29] Raspberry Pi Foundation, "Raspberry Pi Model A+," at 10.03.15. [Online]. Available: <http://www.raspberrypi.org/products/model-a-plus/>
- [30] dresden elektronik, "deRFsam3-23M10," at 10.03.15. [Online]. Available: <http://www.dresden-elektronik.de/funktechnik/products/radio-modules/oem-modules-derfsam3>
- [31] A. Corporation, "Low Power,2.4GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications," Tech. Rep. 8321, oct 2011.
- [32] L. Fu and R. Schwebel, "RT PREEMPT HOWTO," at 10.03.15. [Online]. Available: https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO
- [33] Atmel Cooperation, "IEEE 802.15.4 MAC Software Package - User Guide," Tech. Rep. AVR2025, Jun. 2012. [Online]. Available: <http://www.atmel.com/Images/doc8412.pdf>

Robust Timing Synchronization Preamble for MIMO-OFDM Systems Using Mapped CAZAC Sequences

Ali Rachini

Fabienne Nouvel

Ali Beydoun and Bilal Beydoun

IETR - INSA de Rennes, France

IETR - INSA de Rennes

GET/UL - Lebanese University

GET/UL - Lebanese University

Rennes, France

Hadath, Lebanon

Email: Ali.Rachini@ul.edu.lb fabienne.nouvel@insa-rennes.fr

bilbey@ul.edu.lb

beydoul@yahoo.fr

Abstract—Orthogonal frequency division multiplexing system provides a promising physical layer for 4G and 3GPP LTE systems in terms of efficient use of bandwidth and high data rates, this technology suffers from Inter Symbol Interference and Inter Carrier Interference. On the other hand, multiple input - multiple output system is deployed along with orthogonal frequency division multiplexing in the new 802.11n standard, which offers many advantages over conventional standards such as 802.11g Wireless LAN. The main challenge of such system is the synchronization between the transmitter and the receiver [1]. A bad timing synchronization causes the loss of a lot of information in a MIMO-OFDM system. In this paper, a robust timing synchronization method is proposed for a MIMO-OFDM systems up to 8×8 as well, where N_t is the transmit antennas and N_r is the receive antennas. The proposed method is based on transmit a mapped orthogonal constant amplitude zero auto correlation sequences over different transmit antennas. The simulations results show that the proposed method has high performance to detect the timing synchronization even at very low signal to noise ratio in additive white Gaussian noise and multipath fading Rayleigh channels. Furthermore, simulation results for our proposed method present a robust timing synchronization against existing methods at a low SNR and for MIMO-OFDM system up to 8×8 , which the coarse and fine timing synchronization are done at the same time at each receive antenna due to the orthogonality of different training sequences transmitted over different transmit antennas.

Keywords - MIMO-OFDM system; fine timing synchronization; coarse timing synchronization; CAZAC sequences; compact preamble.

I. INTRODUCTION

The wireless-communications revolution grows continuously in order to increase throughput, which can only be achieved through the development of new communication technologies. In this context, different wireless communication technologies offer enormous increase of channel capacity like Multiple Input Multiple Output - Orthogonal Frequency Division Multiplexing (MIMO-OFDM) systems. Therefore, the combination between MIMO and OFDM systems is proposed in 802.11n [2].

The OFDM [3] is a digital Multi-carrier modulation technology in which a large number of closely spaced orthogonal subcarriers are used to carry the data. OFDM

became a very popular multi-carrier modulation technique for transmission of signals over wireless channels. OFDM has been deployed in many applications like IEEE 802.11a, HIPERLAN/2 wireless LANs, Digital Video Broadcasting, and satellite radio.

It divides the data into several orthogonal and parallel data streams (N_{sc}) called sub-carrier or sub-channel. Each sub-carrier is modulated with a conventional modulation such as M-ary schemes like Phase Shift Keying (M-PSK) or Quadrature Amplitude Modulation (QAM), also, the total data rate is maintained similar to those in a conventional single-carrier modulation scheme in the same bandwidth. To maintain the orthogonality, the space required between two consecutive sub-carriers is $\Delta f = \frac{1}{T_s}$, where T_s is the duration of OFDM symbol.

The implementation of OFDM systems is very easy, on the other hand, the OFDM modulator/demodulator can be done by a simple Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT) algorithm [4], respectively. The main drawback of OFDM technology is high Peak-to-Average Power ratio (PAPR), which means randomly sinusoidal leads occurred during transmission of the OFDM signal.

Otherwise, OFDM technology suffers from Inter Symbol Interference (ISI) and Inter Carrier Interference (ICI). OFDM uses Cyclic Prefix (CP) or Guard Interval (GI) in order to combat the ISI and ICI introduced by the multi-path channel through, which the signal is propagated. The main idea is to append the last part of the OFDM time-domain waveform from the back to the front to create a guard period. The duration of the guard period T_g should be longer than τ_{max} , where τ_{max} designed the Channel Impulse Response (CIR) of the target multi-path environment. The total duration of the OFDM symbol is $T_{tot} = T_s + T_g$.

Furthermore, Multiple-Input Multiple-Output (MIMO) system is an array of N_t transmit antenna and N_r receive antenna. Such systems are used to improve wireless systems capacity, range and reliability. Several applications, based

on MIMO technology, have been proposed in various communication standards as Worldwide Interoperability for Microwave Access (WiMax), evolved High-Speed Packet Access (HSPA+), Wireless Fidelity (WiFi), 3rd and 4th generation of mobile network and Long-Term Evolution (LTE). MIMO system offers a way to increase data throughput and link range without additional bandwidth or increased transmit power. In order to achieve this goal, MIMO system spread the same total transmit power over different transmit antennas to improve the spectral efficiency (Spatial Multiplexing (SM)). On the other hand, MIMO uses Space Time Coding (STC) in order to improve the link reliability.

- 1) Spatial Multiplexing technique (SM): The Spatial multiplexing is a transmission technique in MIMO wireless communication used to transmit independent and separately encoded data streams, from each of the multiple transmit antennas. This technique is used in order to increase the throughput of such wireless communication system. Therefore, the space dimension is reused, or multiplexed, more than one time. If the transmitter is equipped with N_t antennas and the receiver has N_r antennas, Foshini et al. [5] and Telatar [6] have shown that the theoretical capacity of the MIMO channel, with N_t and N_r configuration, grows linearly with $\min(N_t, N_r)$ rather than logarithmically. The channel capacity of a MIMO system is defined by (1) [5] [6]:

$$C = \log_2 \left[\det \left(I_{N_r} + \frac{\rho}{N_t} H H^\dagger \right) \right] \text{bps/Hz.} \quad (1)$$

with

- N_t : Number of transmit antennas.
- N_r : Number of receive antennas.
- I_{N_r} : Identity matrix $N_r \times N_r$.
- $(\cdot)^\dagger$: Conjugate transpose.
- H : MIMO channel matrix $N_t \times N_r$.
- $\rho = \frac{P}{N_o \cdot B}$: Signal to noise ratio (SNR).
- P : Total transmitted power.
- N_0 : Power Spectral Density (PSD).

- 2) Spatial Diversity technique (SD): Spatial diversity technique rely on transmitting simultaneously, redundant copies of data stream on different transmit antennas. The receiver combines the multiple copies of data on each of the received antennas, due to this combination, the error rate of retrieved data will be pretty much less [7]. Space Time Code (STC) is the technique to exploit spatial diversity, which may be split into two main types:

- Space-Time Trellis Codes (STTCs) [8]: This technique is used to distribute a trellis code over multiple transmit antennas and multiple time-slots, furthermore, it provides both coding gain and diversity gain.
- Space-Time Block Codes (STBCs): The STBC is a technique to transmit multiple copies of a data stream across N_t transmit antennas in a MIMO

system. It exploits the spatial diversity and increases the reliability of transmission. This type of code is divided into three main approaches [9]: OSTBC (Orthogonal Space-Time Block Codes), NOSTBC (Non-Orthogonal Space-Time Block Codes) and QSTBC (Quasi-Orthogonal Space-Time Block Codes).

In this paper, we will focus on spatial diversity technique using STBC (Space-Time Block Code) with Alamouti [10] encoder.

The combination of MIMO-OFDM systems are used to reach the higher data rate transmission or improve the spectrum efficiency of wireless link reliability in wireless communication systems. The main challenges of such systems is the synchronization between transmitter and receiver. Two main types of synchronization are necessary, the frequency and the timing synchronization. The frequency synchronization is to correct the phase error caused by the mismatch of the local oscillator (LO) between transmitter and receiver [11] or due to the Doppler effect. On the other hand, Timing synchronization is divided into frame timing synchronization (Coarse timing synchronization) and symbol timing synchronization (Fine timing synchronization). Frame timing synchronization used to detect the arrival of the OFDM frame and symbol timing synchronization is needed in order to detect the beginning of each OFDM symbols on each frame. Here we focus on symbol timing synchronization in MIMO-OFDM systems.

In the literature, several synchronization approaches have been proposed for OFDM and MIMO-OFDM systems [1], [12]–[20]. The main idea is to find a good synchronization preamble, at the transmitter, in order to detect the packet arrival, at the receiver.

In this paper, we propose a robust timing synchronization preamble for MIMO-OFDM systems using orthogonal CAZAC (Constant Amplitude Zero Auto-Correlation) sequences. The CAZAC sequences [21] have constant amplitude and zero autocorrelation for all non-zero shifts. The main characteristics of CAZAC sequences are their correlation functions. They have a good autocorrelation function and their crosscorrelation function is near zero. Due to their orthogonality, CAZAC sequences reduce inter-code interference between multiple antennas and have a lower PAPR. As a result, CAZAC sequences are regarded as optimum preamble for timing synchronization in MIMO-OFDM systems.

This paper is organized as follows. Section II describes the MIMO-OFDM system based on STBC code. Existing approaches and related work are presented in Section III. Section IV presents the criteria for selecting a good synchronization sequences. The working principle of the proposed method and the different preamble structure is presented in Section V. Simulation results and conclusion are discussed in

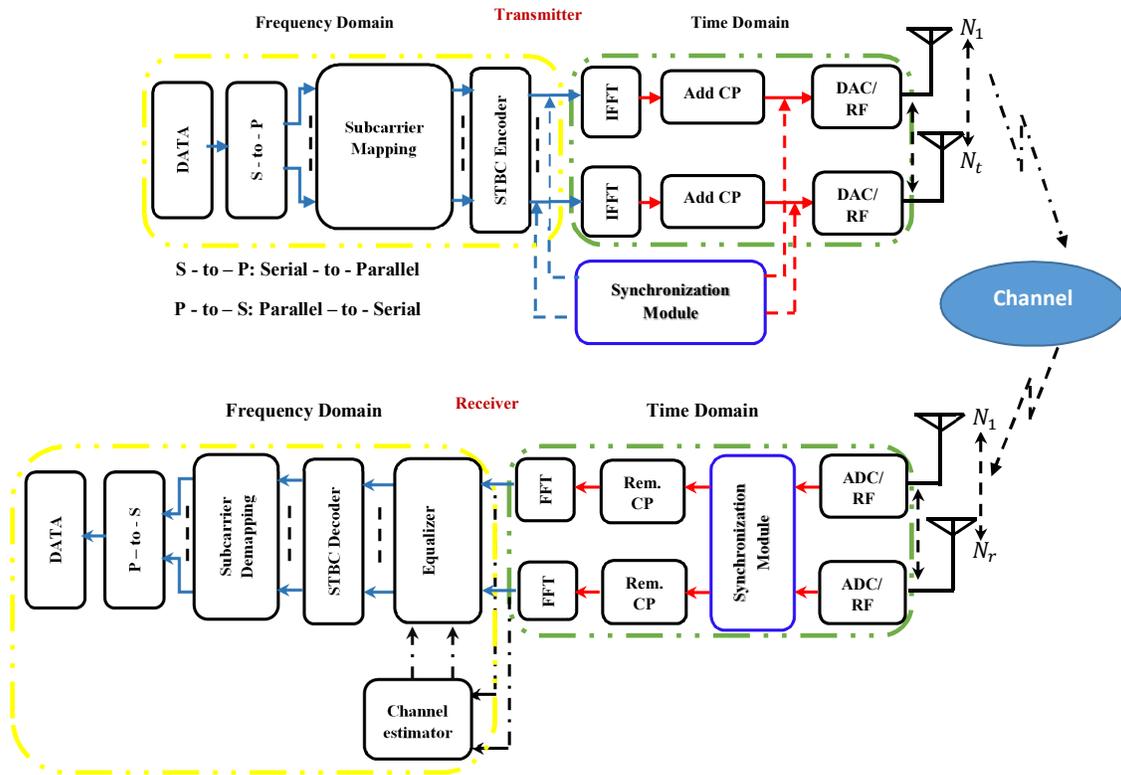


Fig. 1: Block diagram of MIMO-OFDM-STBC transmitter and receiver

Sections VI and VII, respectively.

II. MIMO-OFDM SYSTEM ARCHITECTURE

Basically, MIMO-OFDM radio communication system consists of a transmitter, a channel, and a receiver. In this section, we present the different parts of MIMO-OFDM communications system. The transmitter generates OFDM symbols, which are modulated using M-air modulation, then, they are transmitted over multiple transmit antennas using STBC block [9] [10]. Figure 1 presents a general MIMO-OFDM system model with N_t transmit antennas, N_r receive antennas and N_{sc} subcarriers per transmit antenna.

A. MIMO-OFDM transmitter

The first part of MIMO-OFDM system is the transmitter. In a OFDM transmitter, information data are transmitted blockwise. The first block is a data block where several serial stream of data are generated. Then, a serial to parallel block (S-to-P) converts the serial data stream to parallel data stream. Subcarrier Mapping block is used in order to map parallel data stream to complex symbols. This block uses different constellation mapping either Phase Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM). After mapping, complex symbols are then introduced into a STBC encoder (in this approach we use Alamouti Encoder). Then, we use IFFT to modulate the parallel data stream in order to generate the OFDM symbols over different transmit antennas.

After performing IFFT, the data is again converted into serial stream. A cyclic prefix block named Add CP consists to insert a Cyclic Prefix (CP) or Guard Interval (GI), which is appended at the start of the serial stream. The cyclic Prefix is actually an exact copy of the last part or T_G samples of the data. The purpose of CP is to remove the ISI and channel effects. The synchronization block is used in order to insert the synchronization preamble at the beginning of each OFDM frame. Two different approaches are presented, the synchronization preamble is appended in frequency domain [16] [20] or in time domain [19]. In this paper, we focus on the first approach.

The transmitted OFDM signal s_i on each transmit antenna T_i is given by:

$$s_i(t) = \frac{1}{\sqrt{N_{sc}}} \sum_{k=0}^{N_{sc}-1} \Re \{ x_k e^{j \cdot 2\pi \cdot f_k \cdot t} \} \quad (2)$$

where x_k is the symbol on the frequency f_k .

B. MIMO channel Model

The modelling of a practical MIMO channel includes the transmit vector, receive vector, multi-path channel matrix and Noise. The MIMO channel model between the transmit antenna T_i and receive antenna R_j , where $i \in \{1, N_t\}$ and $j \in \{1, N_r\}$, is given by:

$$H(t) = \sum_{l=1}^L H_l \delta(t - \tau_l) \quad (3)$$

where H_l are the matrix coefficients of the l^{th} path. This matrix is $N_t \times N_r$. δ represents the pulse function and L is the maximum number of multi-paths. H_l is given by:

$$H_l = \begin{bmatrix} h_{1,1}^l & h_{1,2}^l & \dots & h_{1,N_r}^l \\ h_{2,1}^l & h_{2,2}^l & \dots & h_{2,N_r}^l \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_t,1}^l & h_{N_t,2}^l & \dots & h_{N_t,N_r}^l \end{bmatrix} \quad (4)$$

C. MIMO-OFDM Receiver

The second part of MIMO-OFDM system is the receiver. The receiver is exactly the reverse of transmitter. The first block after the analog to digital converter (ADC) is the timing synchronization block. After a good timing synchronization, the cyclic prefix of each OFDM symbol is removed. After removing CP, we perform Fast Fourier Transform (FFT) to return the data back into frequency domain. The data is then fed into the equalizer and channel estimator. After equalization, the data are decoded by STBC decoder. Then, a Subcarrier De-mapping block is presented in order to demodulate and recover the binary information. The parallel to serial (P-to-S) converter allows to reformatting the binary bit stream.

The received signal r_j on each receive antenna R_j is given by:

$$\begin{aligned} r_j(t) &= \sum_{i=1}^{N_t} [h^{i,j}(\tau, t) \star x_i(t)] + n_{ij}(t) \\ &= \frac{1}{\sqrt{N_{sc}}} \sum_{i=1}^{N_t} \sum_{p=1}^{P_{ij}} \left[\alpha_p(t) e^{-j2\pi f_k \tau_p(t)} \star \right. \\ &\quad \left. s_i[\tau - \tau_p(t)] \right] e^{j2\pi f_k t} \\ &\quad + n_{ij}(t) \end{aligned}$$

where h_{ij} is the channel between the transmit antenna T_i and the receive antenna R_j , τ is the propagation delay for the different channels paths, α_p is the attenuation for the p^{th} path, $s_i(t)$ is the OFDM transmitted signal, P_{ij} is the number of path between T_i and R_j and n_{ij} is the Additive white Gaussian noise (AWGN) noise between T_i and R_j .

III. RELATED WORK

In the literature, several synchronization approaches have been proposed for MIMO-OFDM, as shown in Section I. The most of the synchronization methods are preamble based, that means, the header of each OFDM frame contains a known preamble structure. As in [22], authors provide a preamble structure based on Loosely Synchronous (LS) codes for timing and frequency synchronization for a MIMO-OFDM

system. This preamble is used in order to detect the beginning of each received frame. The main characteristics of LS codes is to have a good autocorrelation and cross-correlation functions within certain vicinity of the zero shifts. In this method, the synchronization process is divided into four stages. The first and the second stage are used in order to estimate the coarse timing synchronization and the coarse frequency synchronization, respectively. The third stage is to detect the beginning of each OFDM symbols in each frame and estimate the channel parameters. the fourth step is used for the fine frequency estimation. The main drawback of this method is the structure of preambles, where it is relatively complex. Another disadvantage of this method is the different stage used in order to detect the beginning of frame.

Another approach proposed in [23] based on Orthogonal Variable Spreading Factor (OVSF) for timing synchronization. In this approach, a Multiple Input-Single Output (MISO) systems 2×1 is considered. The length of each OFDM symbol and their CP is 256 and 32, respectively. The synchronization preamble has the same length as the CP is appended at the beginning of each OFDM frame. As result, this approach shows that for MISO-OFDM systems 2×1 , the timing acquisition probability is 1 for an $SNR \geq -5 \text{ dB}$. Here, timing acquisition probability describes the probability to detect the timing synchronization point. The main drawback of this approach is that the synchronization preamble is appended in the time domain. With such hardware implementation, authors need an extra block to insert the preamble in time domain, while, in frequency domain their is no need to this block due to the IFFT.

Based on Schmidl and Cox's approach [12], Farhan proposed in [24] a modified approach that is not using the training sequence and making cyclic prefix (CP) as the reference in order to obtain efficiency in transmitting power. This approach uses the sliding window technique to compute the correlation of the received signal with the cyclic prefix. The main drawback of this method is the correlation with the CP in multipath fading channel. In such channel, when the receiver receives several delayed path with CP, the timing metric obtained by the correlation shows more correlation peak, then the receiver would not able to detect the start of OFDM frame.

A compact preamble design for synchronization in distributed MIMO-OFDM systems has been proposed in [25]. In this approach, a preamble structure based on exclusive sub-band has been proposed. Adjacent sub-bands are spaced by a guard bands to reduce the interference between sub-bands. The total length of the proposed preamble based on CAZAC sequences has the same length as an OFDM symbol. The simulation results shown that for a MISO-OFDM (3×1), the acquisition probability for timing synchronization is 70% for an $SNR = 5 \text{ dB}$. In this work, we compare the simulation results of our proposed approach with those of the method

proposed by Chin-Liang et al. [25].

The proposed method in [25] suffers from several disadvantages, mainly the complexity to generate the preamble structure for a large number of transmit antennas. In [25], when the number of transmit antennas increases, the size of sub-bands should be reduced to take into account all transmit antennas. Therefore, at the receiver, the acquisition probability for timing synchronization decreases due to the length of synchronization sequence.

IV. OPTIMAL TRAINING SEQUENCES FOR TIMING SYNCHRONIZATION

Timing synchronization methods are performed using training sequences at the beginning of each OFDM frame in MIMO-OFDM systems. The main characteristic of training sequences is to have good autocorrelation and cross-correlation properties. At receiver, this characteristic provides a good detection of a correlation peak as closed as possible to a Dirac pulse. The main training sequences used in the state of art are listed below.

A. Gold sequences

Gold sequences [26] of length N are constructed using a preferred pair of Pseudorandom Noise (PN) sequences. Let a and b are two preferred pair of m -sequences, those sequences have a three valued correlation function:

$$\theta_{(a,b)} = -1, -t(m) \text{ or } t(m) - 2 \quad (5)$$

where

$$t(m) = \begin{cases} 1 + 2^{(m+2)/2} & \text{if } m \text{ is even} \\ 1 + 2^{(m+1)/2} & \text{if } m \text{ is odd} \end{cases} \quad (6)$$

The set of Gold sequences includes the preferred pair of m -sequences a and b , and the mod 2 sums of a and cyclic shifts of b represented by the operator T^{-p} . The set S_{gold} of Gold sequences is given by:

$$S_{gold} = \{a, b, a \oplus b, a \oplus T^{-1}b, \dots, a \oplus T^{-(N-1)}b\} \quad (7)$$

The maximum correlation value for any two Gold sequences in the same set is equal to the constant $t(m)$.

The main advantage of Gold sequence lies in sending such sequences in periodic way to retain the good correlation properties. In synchronization, such sequences are used aperiodically, on the other hand, Gold sequences loses their good correlation properties. The autocorrelation and cross-correlation functions of aperiodic Gold sequences are shown in Figure 2, where index represents indices at which the correlation was estimated.

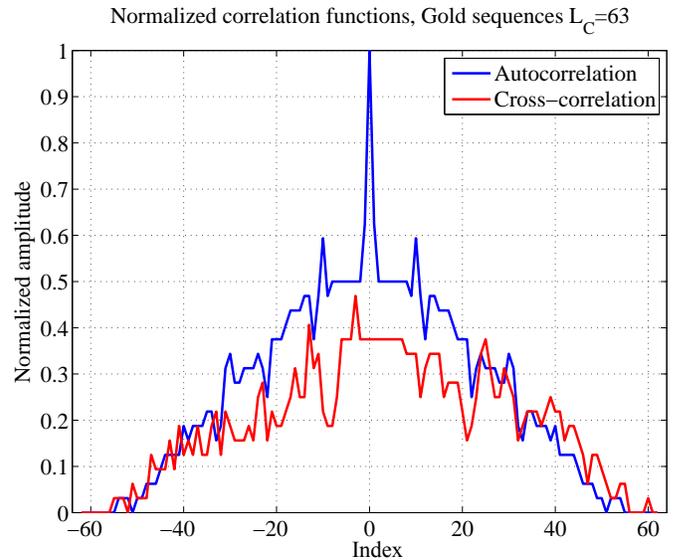


Fig. 2: Autocorrelation and cross-correlation of Gold sequence, $N = 63$

B. Walsh-Hadamard code

Another generation of code called Walsh-Hadamard code. Such codes are orthogonal and built from an initial Hadamard's matrix [27]. Hadamard's matrices are given by:

$$H_1 = [1], H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \dots, H_{2^k} = \begin{bmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{bmatrix} \\ = H_2 \otimes H_{2^{k-1}} \text{ for } 2 \leq k \in \mathbb{N} \quad (8)$$

where \otimes denotes the Kronecker product. An Hadamard matrix H_n satisfies the following property:

$$H_n \cdot H_n^T = nI_n$$

where H_n^T is the conjugate transpose of H_n and I_n is a $n \times n$ identity matrix.

The main advantage of Hadamard code is the orthogonality between the different code. On the other hand, the autocorrelation function for some code has secondary peak as shown in Figure 3.

C. CAZAC sequences

A Constant Amplitude Zero Auto-Correlation (CAZAC) sequence [28] $c(m)$ is a complex sequence and has constant magnitude, $|c(m)| = 1$ for $m \in [0, L_C - 1]$ where $L_C = 2^n$ is the finite length of $c(m)$, $n \in \mathbb{N}$, and has zero-autocorrelation function with shifted version of the same sequence. The cross-correlation function of CAZAC sequences has a value near to zero.

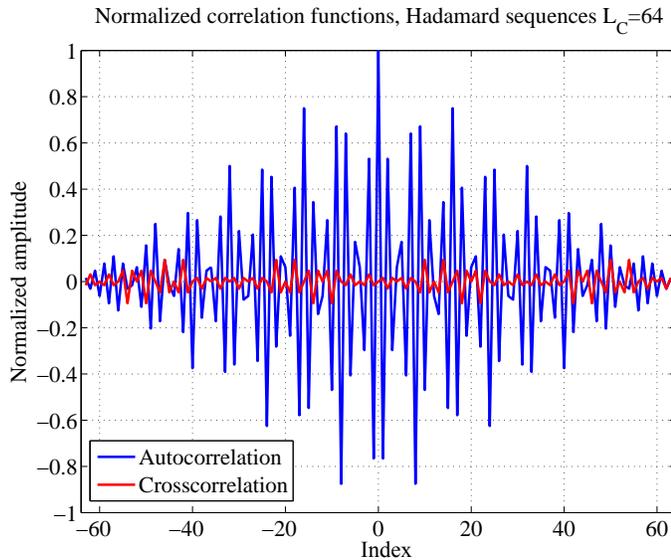


Fig. 3: Autocorrelation and cross-correlation of Hadamard sequence, $L_C = 64$

Let $C(k)$, in frequency domain, be a CAZAC sequence of length L_C , $C(k)$ is given by:

$$C(k) = \begin{cases} e^{j\left(\frac{\pi P k(k+1)}{L_C}\right)} & \text{if } k \text{ is odd} \\ e^{j\left(\frac{\pi P k^2}{L_C}\right)} & \text{if } k \text{ is even} \end{cases} \quad (9)$$

where $P \in \mathbb{N}$ is a prime number with L_C and $k \in \{0, L_C - 1\}$ is the index of the sample.

After IFFT algorithm, the corresponding sequence of $C(k)$ in the time domain ($c(m)$), is given by:

$$c(m) = \frac{1}{L_C} \sum_{k=0}^{L_C-1} C(k) \cdot e^{j\left(\frac{2\pi}{L_C}\right)mk}, m \in [0, L_C - 1] \quad (10)$$

The normalized autocorrelation and cross-correlation functions of CAZAC sequences of length $L_C = 64$ are shown in Figure 4.

D. Optimal sequence selection

In MIMO system, unlike Single Input Single Output (SISO) system, we need to send several data at one time according the number of transmit antennas. In this context, the optimal training sequences should have a good autocorrelation and crosscorrelation functions in order to distinguish the different received signal at the receiver. Gold sequences have a good autocorrelation function, on the other hand, they have a high value for their cross-correlation function. Hadamard sequences have a good autocorrelation and crosscorrelation functions

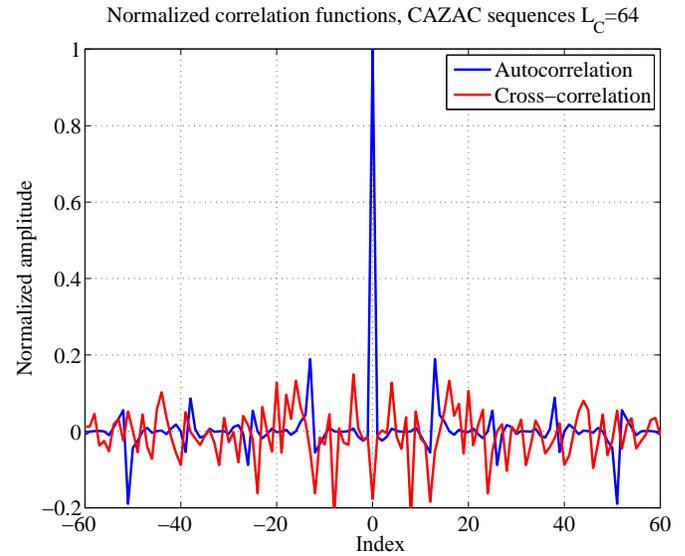


Fig. 4: Autocorrelation and cross-correlation of CAZAC sequence, $L_C = 64$

due to their orthogonality, while, for some sequences, the autocorrelation has secondary correlation peaks. On the other hand, CAZAC sequences show a good autocorrelation and crosscorrelation functions due to their orthogonality and complex value. After a comparison between the characteristics of different sequences, our work was focused on the use of CAZAC sequence as training sequences for timing synchronization in MIMO-OFDM systems.

V. PROPOSED TIMING SYNCHRONIZATION PREAMBLE

In this section, based on [1], we propose our robust timing synchronization preamble in MIMO-OFDM systems based on CAZAC sequence. Let C be a CAZAC sequence of length L_C , where L_C represents the size of synchronization preamble divided by 2, in other term $L_C = L_{FFT}/2$ where L_{FFT} is the size of the FFT, and C^* denotes the conjugate of C . We propose in this section two different structures, as follows.

A. First preamble structure

The timing synchronization preamble is generated in frequency domain, then it is added at the beginning of each OFDM frame according to transmit antenna. Figure 5 shows the structure of the first proposed preamble in frequency domain over different transmit antennas.

The preamble structure in Figure 5 relies in sending a CAZAC sequence (C) over the odd subcarrier, in frequency domain, and C^* on the even subcarrier. The preamble ϕ^i that transmitted on the i^{th} transmit antenna is given by the following equation:

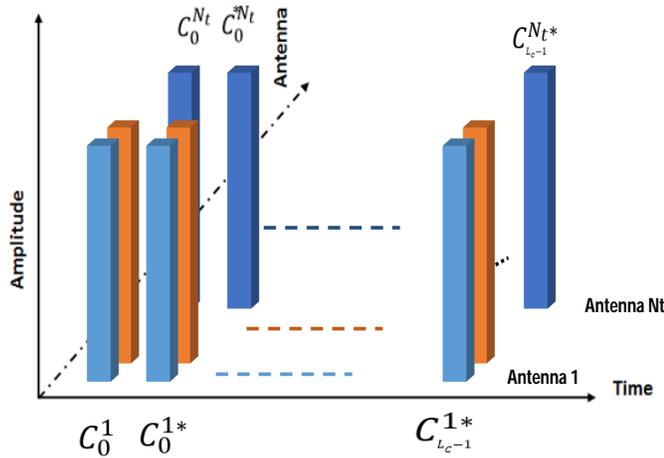


Fig. 5: First preamble structure in frequency domain over different transmit antennas

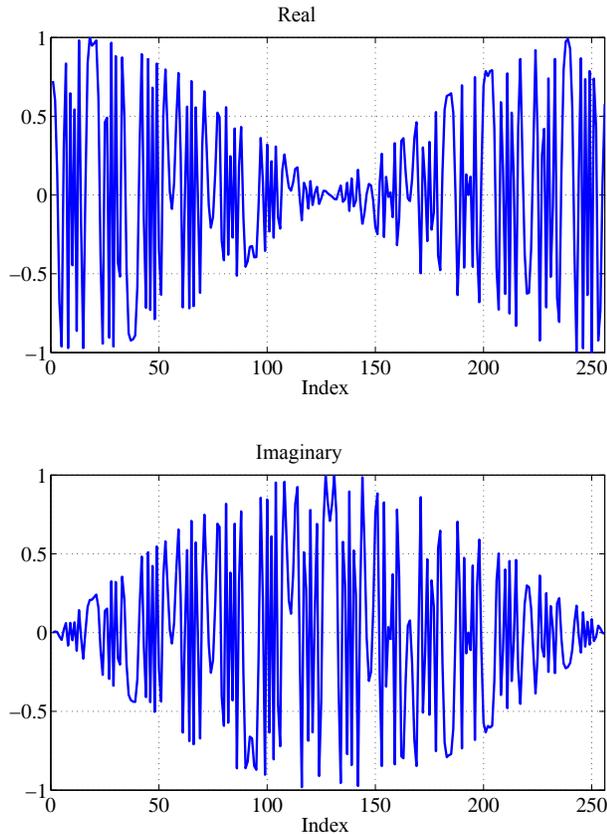


Fig. 6: Real and imaginary parts of the first preamble structure in time domain

$$\varphi^i(k) = \begin{cases} C^i\left(\frac{k}{2}\right) & \text{if } k \bmod 2 = 0 \\ C^{i*}\left(\frac{k-1}{2}\right) & \text{if } k \bmod 2 \neq 0 \end{cases} \quad (11)$$

where $k \in \{0, L_{FFT} - 1\}$ and $L_{FFT} = 2 \cdot L_C$.

The term C_k^i is the sample of the CAZAC sequence carried by the k^{th} subcarrier and transmitted by the transmitting antenna T_i . The proposed method can be applied regardless of the number of transmit or receive antennas.

Figure 6 shows the real and imaginary parts of timing synchronization preamble in time domain.

The combination of a CAZAC sequence C with its conjugate C^* gives a time-domain complex envelope form that have a good autocorrelation and cross-correlation functions. This combination does not destroy the orthogonality between subcarriers, and it retains the orthogonality between different preambles over different transmit antennas.

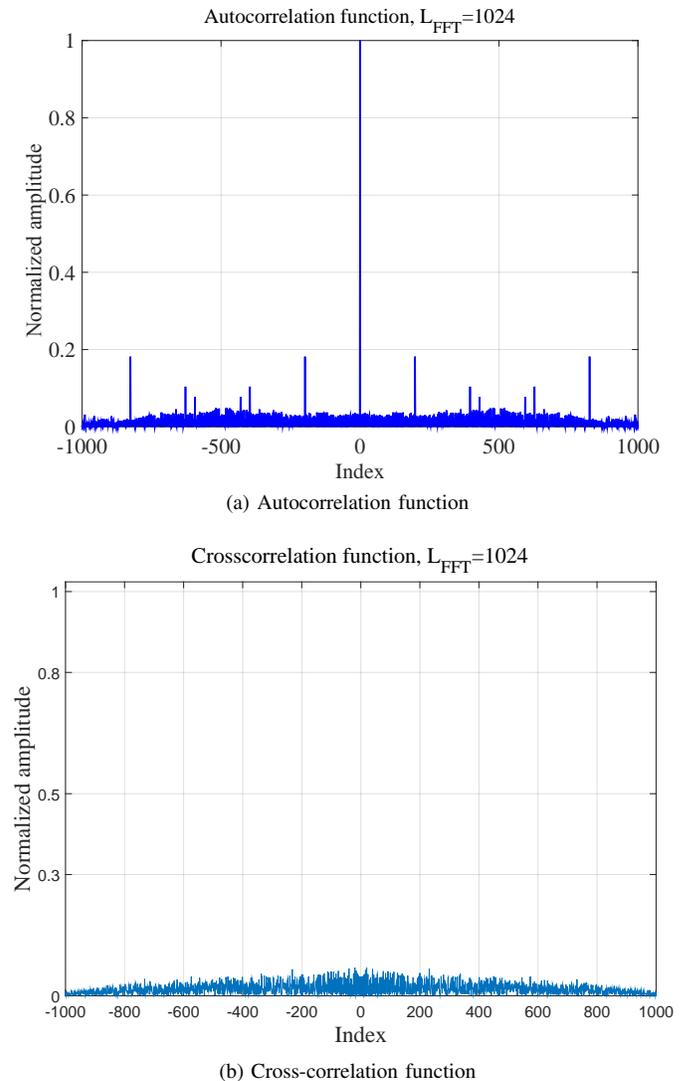


Fig. 7: Autocorrelation and cross-correlation functions of the first preamble structure ($L_{FFT} = 1024$)

Figure 7 presents the autocorrelation (Figure 7a) and the

cross-correlation (Figure 7b) functions of the first preamble structure. This preamble shows a good correlation functions in order to detect the timing synchronization peak.

B. Second preamble structure

The second preamble structure consists of dividing the preamble into two parts of length $L_C = L_{FFT}/2$ each one. The first part contains an entire CAZAC sequence C of length L_C , while, the second part contains the conjugate of C denoted C^* . Figure 8 presents the preamble structure over different transmit antennas in frequency domain.

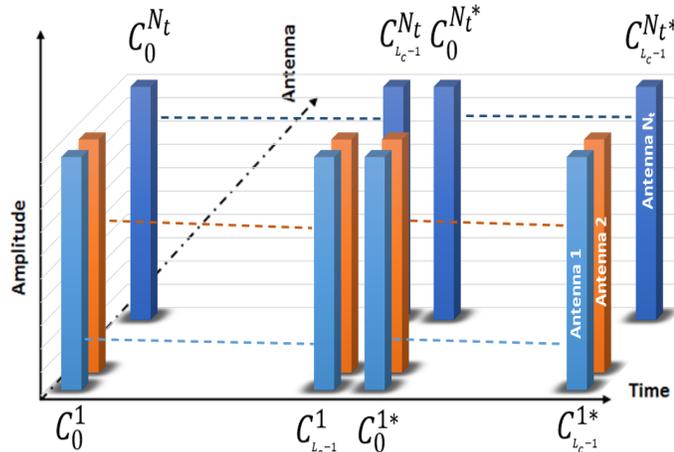


Fig. 8: Second preamble structure in frequency domain over different transmit antennas

Figure 9 presents the autocorrelation and cross-correlation functions of the second preamble structure in time domain.

Let φ^i be the preamble sent on the i^{th} transmit antenna, the equation of this preamble in frequency domain is given by:

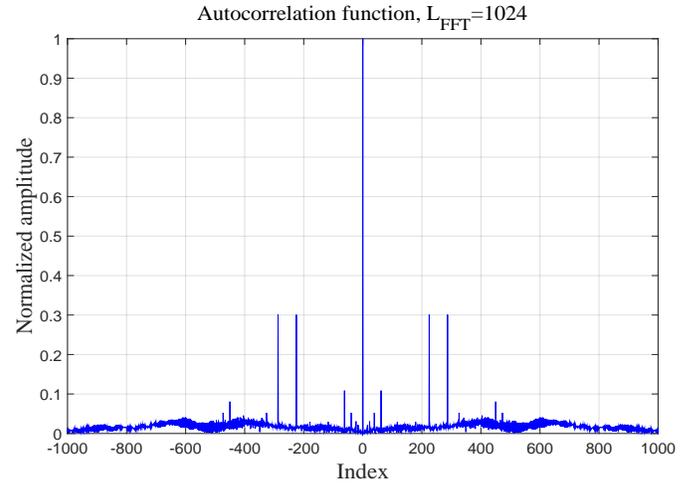
$$\varphi^i(k) = \begin{cases} C^i(k) & \text{if } 0 \leq k < L_C \\ C^{i*}(k - L_C) & \text{if } L_C \leq k < L_{FFT} \end{cases} \quad (12)$$

VI. SIMULATIONS RESULTS

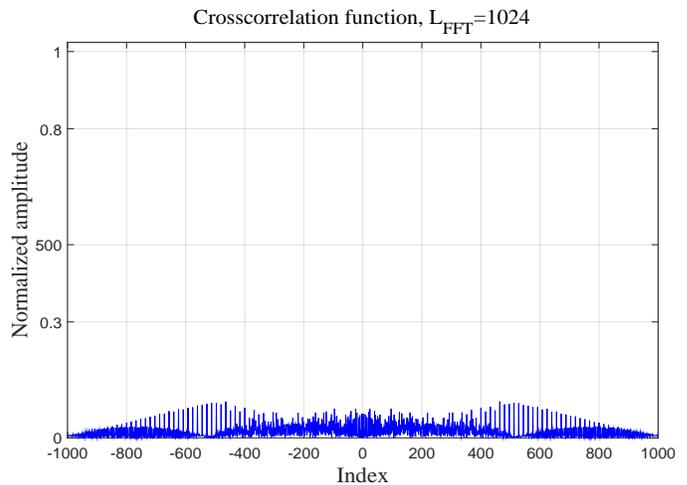
In this section, we present the simulation parameters and simulation results for different preamble's structure in AWGN channel and multipaths fading channel, in order to evaluate the performance of our proposed preambles against [25].

A. System specifications & simulation parameters

In order to improve the simulation results, we simulate our preamble structures with different system specification and simulation parameters. Simulations results are done with SISO-OFDM and MIMO-OFDM systems up to 8×8 . On the other hand, the OFDM system consists of 512 and 1024 subcarriers, where $L_{FFT} = \{512, 1024\}$, respectively. The channel model was considered as Rayleigh multipath



(a) Autocorrelation function



(b) Cross-correlation function

Fig. 9: Autocorrelation and cross-correlation functions of the second preamble structure ($L_{FFT} = 1024$)

fading channel with 6 paths sample-spaced with T_s , where T_s describes the sampling time; this channel is suggested by the IEEE 802.11 Working Group [29]. Other simulation parameters are summarized in Tables I and II.

TABLE I: System Specifications and Requirements

Parameters	Justification	
	Value	Description
System	8×8	SISO and MIMO-OFDM system up to 8×8
L_{FFT}	1024 & 512	Length of IFFT/IFFT
L_{CP}	$L_{FFT}/4$	Length of Cyclic Prefix
Sequences	CAZAC	Type of synchronization sequences
L_C	$L_{FFT}/2$	Length of synchronization sequences
SNR in dB	0 to 25	SNR over all the OFDM Frame

B. Timing synchronization algorithm

The main drawback of the most of timing synchronization algorithm is the complexity of frames and symbols detection.

TABLE II: Power profile and channel model

Simulation Parameters	Value
Channel Type	Multi-path Rayleigh and AWGN channel
Number of channel taps between different antennas	6
Propagation delay between different multipath	$[0.T_s, 1.T_s, 2.T_s, 3.T_s, 4.T_s, 5.T_s]$
The power of each multipath	$[0.8111, 0.1532, 0.0289, 0.0055, 0.0010, 0.0002]$

Our proposed method consists of detecting coarse and fine timing synchronization in one operation, this is the main advantage of our proposed method. We implement at each receiver a correlation function \mathcal{R}_{r_j, seq_j} in order to detect the timing synchronization peak between the received signal r_j and the local sequence seq_j generated by the receive antenna R_j . This correlation is done in time domain. Due to the mapped CAZAC sequence (C and C^*) in each preamble, the correlation between received signal and local sequence may give a high peak's value, this function is calculated as following:

$$\mathcal{R}_{r_j, seq_j}(n) = \sum_{n=0}^{L_{FFT}-1} [r_j(n) * seq_j(n - \tau)] \quad (13)$$

where n is the index of the sample. The correlation function \mathcal{R}_{r_j, seq_j} feed into an estimator in order to detect the coarse timing synchronisation. The timing synchronization estimate is given by:

$$\hat{ind}_n = \underset{n}{\operatorname{argmax}} \{ \|\mathcal{R}_{r_j, seq_j}(n)\| \} \quad (14)$$

where n is considered as the coarse timing synchronization point. At the same time, and, by shifting the FFT window, we can find the fine timing synchronization or the beginning of each OFDM symbol on each frame. Let P_{SYNC} describes the timing synchronization acquisition probability. P_{SYNC} presents the probability of successful timing synchronization at receiver.

C. Simulation results for the first preamble structure

Simulation results for all preamble structures, are done using simulation parameters in Tables I and II. Figures 10 and 11 show the acquisition probability for different SISO and MIMO-OFDM systems, where the length of preamble are $L_{FFT} = 1024$ and $L_{FFT} = 512$.

Figure 10 presents a good timing synchronization for a low SNR. For an $SNR = -5dB$, the $P_{SYNC} \geq 90\%$ for all MIMO-OFDM system up to 8×8 . Therefore, for an $SNR = 0dB$, the proposed timing synchronization preamble shows a perfect timing synchronization for SISO-OFDM system. The $P_{SYNC} \geq 97\%$ for MIMO-OFDM system 2×2 for the same SNR. For a MIMO-OFDM system 4×4 the $P_{SYNC} \geq 96\%$ at an $SNR = 5dB$. On the other hand, for MIMO-OFDM system 8×8 , the acquisition probability

P_{SYNC} reaches 98% at an $SNR = 10dB$.

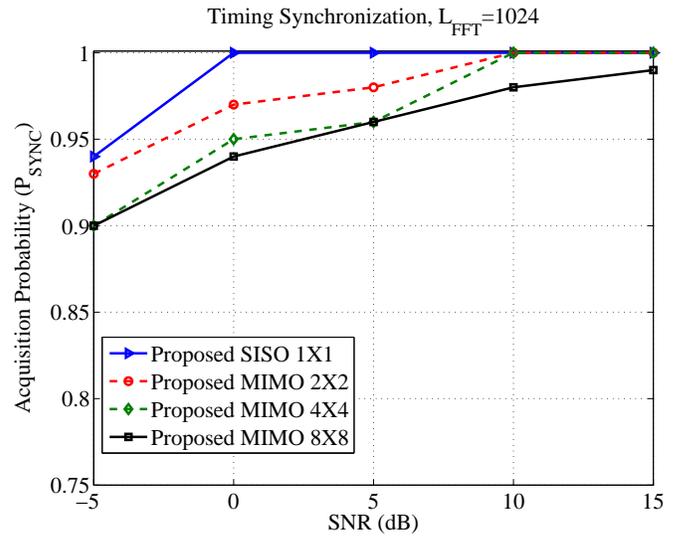


Fig. 10: Timing synchronization performance of the first proposed preamble with $L_{FFT} = 1024$

Figure 11 presents the performance of our synchronization preamble of length $L_{FFT} = 512$. In this figure, the acquisition probability P_{SYNC} is greater than 97% for both SISO-OFDM and MIMO-OFDM 2×2 systems at an $SNR = 0dB$. Therefore, $P_{SYNC} \geq 90\%$ for MIMO-OFDM 4×4 system at an $SNR = 0dB$. On the other hand, the P_{SYNC} reaches 80% at an $SNR = 5dB$ for MIMO-OFDM system 8×8 .

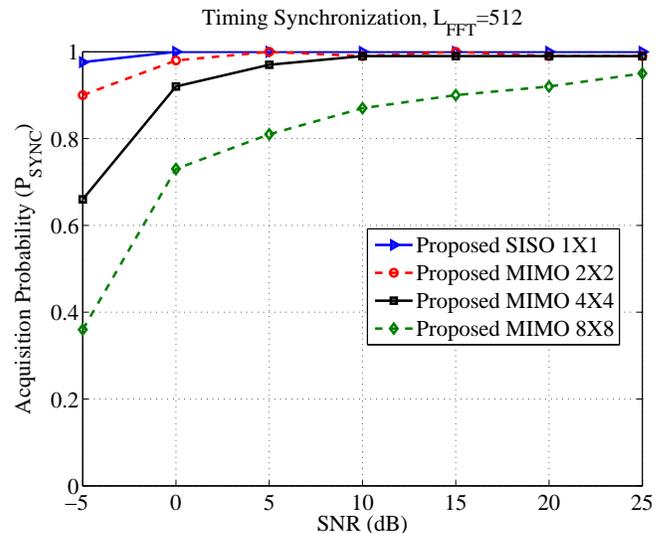


Fig. 11: Timing synchronization performance of the first proposed preamble with $L_{FFT} = 512$

Table III summarizes the simulation results of Figures 10 and 11. It can be shown that the performance of our timing synchronization method increases with the length of L_{FFT} .

TABLE III: Comparison between the acquisition probability of different MIMO-OFDM systems, in term of SNR and length of FFT

MIMO-OFDM system	Acquisition probability		
	P_{SYNC}	SNR (dB)	L_{FFT}
MIMO-OFDM 2x2	$\geq 97\%$	> 0 dB	1024
	$\geq 96\%$	> 0 dB	512
MIMO-OFDM 4x4	$\geq 95\%$	> 0 dB	1024
	$\geq 93\%$	> 0 dB	512
MIMO-OFDM 8x8	$\geq 94\%$	> 0 dB	1024
	$\geq 78\%$	> 0 dB	512

Moreover, the results of Figure 10 ($L_{FFT} = 1024$) show a good performance against those presented in Figure 11 ($L_{FFT} = 512$).

In order to evaluate the performance of our proposed method, we conducted an extensive comparison of our approach with the synchronization scheme of [25]. Hung and Chin Wang [25] used a subband-based preamble based on CAZAC sequences. The main drawback of this method is the number of transmit antennas. As the number of transmit antennas increases, the length of synchronization sequence, on each transmit antenna, decreases. Therefore, the value of the synchronization peak at the receiver decreases.

Figure 12 presents the performance between our proposed approach and the synchronization scheme of [25]. Simulation results in Figure 12 are done with the simulation parameters of Tables I and II with a synchronization preamble of length $L_{FFT} = 256$, and MIMO-OFDM system 2×2 and 3×3 .

Simulation results of our proposed approach have a good performance against [25] at a low SNR. The acquisition probability P_{SYNC} for our method is greater than 90% at an $SNR \geq 5$ dB for both MIMO-OFDM 2×2 and 3×3 system. Therefore, the proposed method in [25] shows that the acquisition probability is between 0.5 and 0.75 at the same value of SNR.

D. Simulation results for the second preamble structure

This section presents the simulation results for the second preamble structure. Simulation results are done performed using the simulation parameters in Tables I and II. The acquisition probabilities (P_{SYNC}) for different length of synchronization preamble ($L_{FFT} = 1024$ and $L_{FFT} = 512$) are shown in Figures 13 and 14, respectively.

Figure 13 shows that for a $L_{FFT} = 1024$ and a $SNR \geq -5$ dB, the acquisition probability P_{SYNC} is greater than 95% for both SISO-OFDM and MIMO-OFDM 2×2 systems. Otherwise, both systems have a perfect P_{SYNC} for

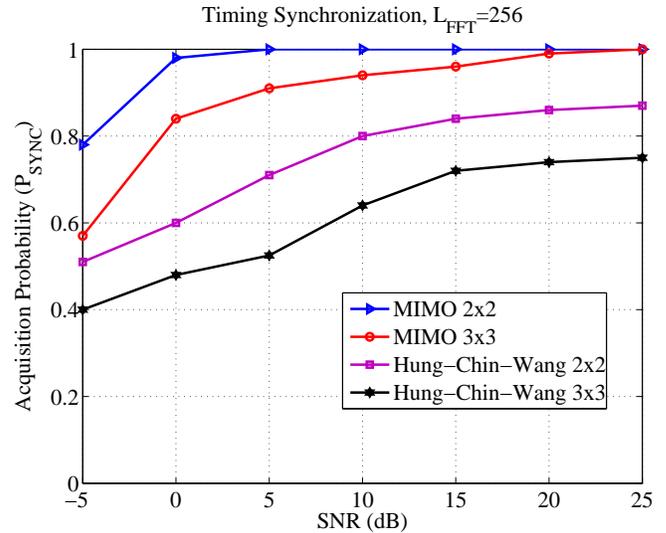


Fig. 12: Comparisons between the proposed approach and subband-based preamble [25]

an $SNR \geq 5$ dB. On the other hand, MIMO-OFDM 4×4 system has a P_{SYNC} greater than 94% for a $SNR \geq 0$ dB, this system has a perfect P_{SYNC} for a $SNR \geq 15$ dB. Therefore, the acquisition probability for MIMO-OFDM 8×8 system, reaches 90% for an $SNR > 2$ dB.

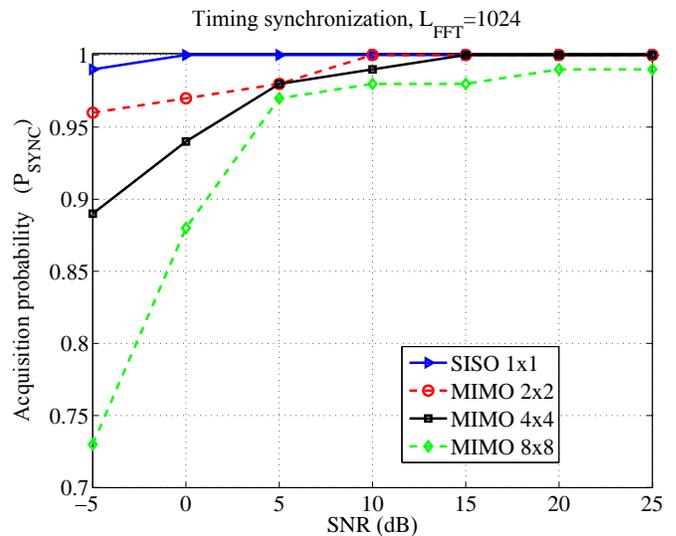


Fig. 13: Timing synchronization performance of the second proposed approach ($L_{FFT} = 1024$)

Figure 14 presents the performance of timing synchronization method for a $L_{FFT} = 512$. As shown in this figure, at an $SNR = 0$ dB, both SISO-OFDM and MIMO-OFDM 2×2 systems have the acquisition probability $P_{SYNC} > 95\%$, and $P_{SYNC} > 90\%$ for MIMO-OFDM 4×4 system. Furthermore, for MIMO-OFDM 8×8 system the P_{SYNC} reaches 70% for

the same SNR .

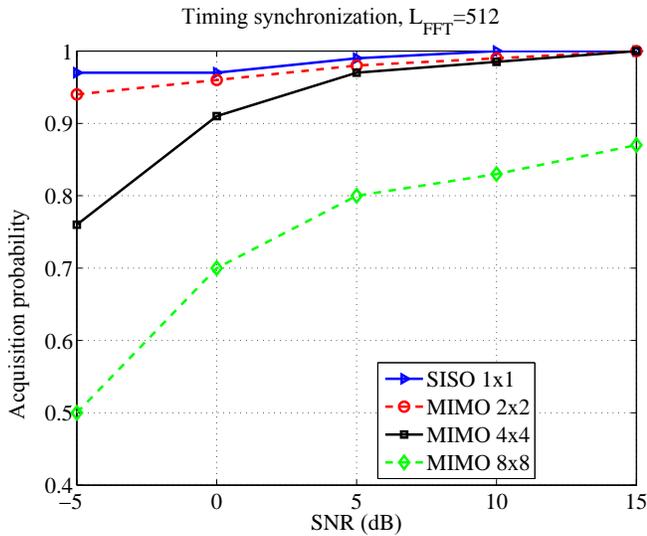


Fig. 14: Timing synchronization performance of the second proposed approach ($L_{FFT} = 512$)

In order to show the performance of our approach clearly, simulation results of our approach are compared with the method proposed in [25], using the same simulation parameters of Tables I and II. The comparison results are shown in Figure 15, where the preamble size is $L_{FFT} = 256$. As shown in this figure, the timing synchronization acquisition probability for our proposed approach is better than the proposed method in [25] even for a low SNR .

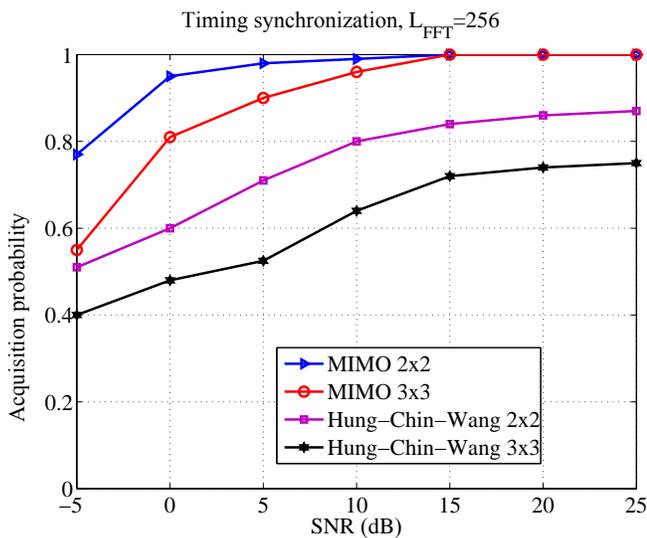


Fig. 15: Comparisons between the second proposed approach and subband-based preamble proposed in [25]

For an $SNR = 0$ dB, for both MIMO-OFDM systems 2×2 and 3×3 , our proposed approach has a $P_{SYNC} > 0.8$, while for Hung-Chin's method the P_{SYNC} not exceed 0.6 for both systems.

VII. CONCLUSION

In the last year, the telecommunications have been growing in order to present a good quality of services (QoS) and large bandwidth. Furthermore, in order to increase the capacity of channel, or to improve the quality of the link, MIMO-OFDM system was presented. On the other hand, such system has a big challenge, which is the timing synchronization. Timing synchronization means how to detect the beginning of each received frames and each symbols in the frame. In order to detect the timing frame synchronization, we proposed two robust timing synchronization preamble. At the transmitter, a synchronization preamble is appended at the beginning of each OFDM frame. This preamble is based on CAZAC sequences, where those sequences have a good autocorrelation and cross-correlation functions.

At each receiver, the received signal correlated with a local sequence generated by the receiver. Due to the mapped of the orthogonal CAZAC sequences over different subcarriers, a correlation peak will appear in order to detect the beginning of frame. In comparison to the subband preamble based proposed by [25], our timing synchronization approaches present a better timing frame synchronization at a low SNR . Finally, we can perform coarse and fine timing synchronization using the same correlation peak.

In this paper, we can find also a few degradation of performance between our two approach. This degradation is due to the mapped CAZAC sequence on each preamble structure. Hence, as future work, it will be interesting to see the performance of our approach for frequency synchronization.

ACKNOWLEDGMENT

This work was supported by the GET of Lebanese University and IETR of Rennes-France.

REFERENCES

- [1] A. Rachini, A. Beydoun, F. Nouvel, and B. Beydoun, "Timing synchronization method for mimo-ofdm systems with cazac sequences," in INNOV 2014, The Third International Conference on Communications, Computation, Networks and Technologies, 2014, pp. 30–35.
- [2] IEEE Standard 802.11n: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, Institute of Electrical and Electronics Engineers, Oct. 2009.
- [3] L. Cimini, "Analysis and simulation of a digital mobile channel using orthogonal frequency division multiplexing," Communications, IEEE Transactions on, vol. 33, no. 7, July 1985, pp. 665–675.
- [4] S. G. Johnson and M. Frigo, "Implementing ffts in practice," 2009.
- [5] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," Wireless Personal Communications, vol. 6, 1998, pp. 311–335.
- [6] I. E. Telatar, "Capacity of multi-antenna gaussian channels," European Transactions On Telecommunications, vol. 10, no. 6, Dec. 1999, pp. 585–595.
- [7] I. Khan, S. A. K. Tanoli, and N. Rajatheva, "Capacity and performance analysis of space-time block coded mimo-ofdm systems over rician fading channel," International Journal of Electrical and Computer Engineering, 2009.

- [8] V. Tarokh, N. Seshadri, and A. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *Information Theory, IEEE Transactions on*, vol. 44, no. 2, 1998, pp. 744–765.
- [9] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block codes from orthogonal designs," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, 1999, pp. 1456–1467.
- [10] S. Alamouti, "A simple transmit diversity technique for wireless communications," vol. 16, no. 8, Oct. 1998, pp. 1451–1458.
- [11] V. Tarokh, A. Naguib, N. Seshadri, and A. Calderbank, "Space-time codes for high data rate wireless communication: performance criteria in the presence of channel estimation errors, mobility, and multiple paths," *Communications, IEEE Transactions on*, vol. 47, no. 2, Feb. 1999, pp. 199–207.
- [12] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for ofdm," *Communications, IEEE Transactions on*, vol. 45, no. 12, Dec. 1997, pp. 1613–1621.
- [13] A. Rachini, A. Beydoun, F. Nouvel, and B. Beydoun, "Timing synchronization of mimo-ofdm systems," in *LAAS 2014, The 20th International Science Conference*. LAAS, 2014, pp. 149–150.
- [14] U. Rohrs and L. Linde, "Some unique properties and applications of perfect squares minimum phase cazac sequences," in *Communications and Signal Processing, 1992. COMSIG '92., Proceedings of the 1992 South African Symposium on*, Sept. 1992, pp. 155–160.
- [15] B. Yang, K. Letaief, R. Cheng, and Z. Cao, "Burst frame synchronization for ofdm transmission in multipath fading links," in *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, vol. 1, 1999, pp. 300–304 vol.1.
- [16] A. Rachini, A. Beydoun, F. Nouvel, and B. Beydoun, "A novel compact preamble structure for timing synchronization in mimo-ofdm systems using cazac sequences," in *INNOV 2013, The Second International Conference on Communications, Computation, Networks and Technologies*, 2013, pp. 1–6.
- [17] J.-J. van de Beek, M. Sandell, M. Isaksson, and P. Ola Borjesson, "Low-complex frame synchronization in ofdm systems," in *Universal Personal Communications. 1995. Record., 1995 Fourth IEEE International Conference on*, Nov. 1995, pp. 982–986.
- [18] L. Li and P. Zhou, "Synchronization for b3g mimo ofdm in dl initial acquisition by cazac sequence," in *Communications, Circuits and Systems Proceedings, 2006 International Conference on*, vol. 2, Jun. 2006, pp. 1035–1039.
- [19] A. Rachini, A. Beydoun, F. Nouvel, and B. Beydoun, "Timing synchronisation method for mimo-ofdm system using orthogonal preamble," in *Telecommunications (ICT), 2012 19th International Conference on*. IEEE, 2012, pp. 1–5.
- [20] A. Rachini, F. Nouvel, A. Beydoun, and B. Beydoun, "A novel timing synchronization method for mimo-ofdm systems," *International Journal On Advances in Telecommunications*, vol. 7, no. 1 and 2, 2014, pp. 22–33.
- [21] R. Frank, S. Zadoff, and R. Heimiller, "Phase shift pulse codes with good periodic correlation properties (corresp.)," *Information Theory, IRE Transactions on*, vol. 8, no. 6, Oct. 1962, pp. 381–382.
- [22] W. Jian, L. Jianguo, and D. Li, "Synchronization for mimo ofdm systems with loosely synchronous (ls) codes," in *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, Sept. 2007, pp. 254–258.
- [23] G. Gaurshetti and S. Khobragade, "Orthogonal cyclic prefix for time synchronization in mimo-ofdm," *International Journal of Advanced Electrical and Electronics Engineering (IJAEED)*, vol. 2, 2013, pp. 81–85.
- [24] F. Farhan, "Study of timing synchronization in MIMO-OFDM systems using DVB-T," *CoRR*, vol. abs/1404.7843, 2014. [Online]. Available: <http://arxiv.org/abs/1404.7843>
- [25] H.-C. Wang and C.-L. Wang, "A compact preamble design for synchronization in distributed mimo ofdm systems," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, Sept. 2011, pp. 1–4.
- [26] R. Gold, "Optimal binary sequences for spread spectrum multiplexing (corresp.)," *Information Theory, IEEE Transactions on*, vol. 13, no. 4, Oct. 1967, pp. 619–621.
- [27] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.
- [28] R. Frank, S. Zadoff, and R. Heimiller, "Phase shift pulse codes with good periodic correlation properties (corresp.)," *Information Theory, IRE Transactions on*, vol. 8, no. 6, Oct. 1962, pp. 381–382.
- [29] B. O'Hara and A. Petrick, *The IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network IEEE Press, 1999.

Context-, Resource-, and User-Aware Provision of Services on Mobile Devices

Andre Ebert, Florian Dorfmeister, Marco Maier, and Claudia Linnhoff-Popien
 Mobile and Distributed Systems Group
 Ludwig-Maximilians-University
 Munich, Germany
 Email: {andre.ebert, florian.dorfmeister, marco.maier, linnhoff}@ifi.lmu.de

Abstract—Today’s smartphones are equipped with numerous different sensors and are capable of providing a large number of diverse services. But due to the high energy consumption of built-in sensors, as well as because of the energy consumed while analyzing gathered raw data, a bottleneck in resource supply is likely to occur during a service provision. This bottleneck leads to one of the biggest challenges regarding the developments on mobile devices: the trade-off between a high short-term service performance and sustainable energy management. Furthermore, despite of numerous hardware improvements (e.g., energy saving displays, batteries with bigger capacities, etc.) this issue remains unsolved. Hence, software-based approaches can be used to optimize the resource and energy management on mobile devices according to the user’s preferences, existing context information, and the current energetic state of a device. Moreover, a holistic energy management enables the system to provide context dependent services.

In this article, we present a concept for custom tailored service provision on mobile devices in combination with EMMA (Energy Management Middleware Architecture), a modular architecture for managing a mobile device’s service infrastructure in relation to its current resource state as well as to the user’s individual preferences. Additionally, we give an insight into our prototypical application, which demonstrates EMMA’s core concepts including its featured approach of individual service provision.

Keywords—Mobile resource management; User-centric service provision

I. INTRODUCTION

Depending on the usage intensity, the batteries of today’s smartphones do not often last longer than for one or two days. Reasons for that are high performance hardware components leading to high energy consumption, extensive sensor usage for data acquisition, improperly developed applications, or others. With this being a known problem, several hardware- and software-based approaches for optimizing the energy consumption on mobile devices have been developed. However, typically, only single components of a mobile system are subject of optimization, while there are no known approaches covering all existing subdomains and the corresponding possibilities for saving energy. Furthermore, due to the fact that most services on a mobile device can be provided by more than one specific procedure, opportunities for using a granular performance and quality concept are facilitated. Disregarding known hardware-based improvements, this leads to the necessity of

an integrated, software-based energy and service management approach in order to protect the limited resources on a mobile device.

But improving a device’s energy consumption is not only about extending its batteries’ life span. Additional goals are the increase of the user’s usage experience and the usability of an energy management system by adjusting specific services to individual user needs and requirements. In order to achieve these objectives, a user’s preferences, current context information and the device’s energetic system state, as well as currently active services must be considered [1]. Furthermore, an energy management system must be able to identify and prioritize services and functionalities which are important or critical to the user. At its best, such a system is able to present these services to the user with a dynamically tailored Quality of Service (QoS) whenever they are requested. QoS is described by individual parameters for each service, depending on its specific output. E.g., the QoS of a data transfer service could be described by its transfer speed and the amount of data to be transferred, the one of a positioning service by its accuracy and the time span needed for acquiring a location.

In order to describe the user’s context and to be able to react to a present situation, context modeling is required first. According to Dey, “Context is any information, that can be used to characterize the situation of an entity [2]. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.” Relying on this definition, it is possible to identify the kind of raw data which is needed for context modeling and to specify sources and procedures for its acquirement. In the following, we present four categories of sources for determining raw context data.

- 1) **Conventional Sensors:** Data about the users environment, specific activities or current whereabouts can be acquired with built-in smartphone sensors such as Global Positioning System (GPS), gyroscope or accelerometer.
- 2) **Communication Interfaces:** Network technologies like Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS) or Long Term Evolution (LTE) plus Bluetooth and Wireless Local Area Network (WLAN) can also be used for determining the user’s location in different granularities [3][4].

- 3) **Media Data Analysis:** Data like taken pictures or surrounding noises can also be analyzed in order to extract information about the user's situation, e.g., concerning current moods or a visited location [5].
- 4) **User Data Mining:** Personal user data like emails, played media files or calendar entries could be valuable for the process of extracting context information. Its analysis can provide knowledge about whereabouts, important dates like birthdays, or upcoming appointments.

However, using some of the presented approaches for the acquisition of context information can be very costly. As shown in Section II, all of them consume energy during the process of raw data gathering, as well as for the procedure of extracting information from it. Consequently, the energy consumption caused by context acquisition always has to be compared to the savings enabled through context-aware energy allocation. Besides, there are also privacy issues to be considered. Especially in 3) and 4), sensitive user data, which may need additional protection by privacy policies, is used for analysis.

We are not aware of any concept for an integrated energy management system, which satisfies all of the aforementioned needs. However, there are a lot of ideas for partial improvements of single components. In the following, we specify the requirements for a holistic system and introduce EMMA, an energy management middleware architecture which considers the dynamic and modular integration of existing energy improvement concepts as well as controlling the device's service provision [1]. Furthermore, it monitors the system's energy status and adapts it according to the current context, active services and the user preferences.

In this section, we presented an introduction to energy management matters on mobile devices as well as pointing out the necessity of providing custom tailored and context-dependent services to the user. In Section II, we provide a brief analysis of a smartphone's energy consumption in order to understand which components are responsible for draining its battery and where improvements can be achieved. Based on a literature survey and existing approaches for energy management on mobile devices, as well as on context-aware mechanisms for optimizing energy consumption, we present the requirements for an extensive context-aware and user-centric mobile energy management architecture in Section III. Eventually, the concept and implementation of this approach, coined Energy Management Middleware Architecture (EMMA), will be described in Section IV. A particular emphasis is laid on the provision of requested services in dependence of the system's current resource state as well as on the user's personal preferences and performance claims. After offering some insights into a prototypical implementation in Section V, a conclusion and future work are addressed in Section VI.

II. RELATED WORK

In order to optimize the energy consumption on a mobile device, as well as using information about its context and energetic state for custom-tailored service provision, one first needs to understand which of a device's components or applications is draining most of its resources. Later on, it is

possible to contrive ideas and optimization concepts based on this knowledge, as well as on already existing optimization and provisioning approaches.

A. Individual measuring of energy consumption

There are two different techniques for measuring energy consumption on smartphones. One relies on software based functions provided by the operating system; the other one uses an external power meter. Manweiler et al. describe the installation and usage of a Monsoon Power Monitor for external power consumption measurement [6]. However, this approach does not pay attention to the individual consumption of the device's components since it only identifies the overall consumption. Examples for software based and combined measurement concepts are eProf [7], PowerTutor [8], and WattsOn [9]. They all make use of predefined energy profiles describing the energy consumption of specific device components and indicate, that network and sensor usage, CPUs, displays and media playback are the main consumers of energy. Additionally, Pathak et al. [7] examined the energy consumption of different popular applications like Facebook, Angry Birds and others and figured out that 65-75% of the consumed energy is used by third-party advertisement modules. Furthermore, they name typical bugs in operation systems, which are responsible for the dissipation of energy.

B. Optimization concepts

In the following, we review different optimization concepts concerning sensors and data transfer. Furthermore, the usage of prediction and data mining in order to prioritize, schedule and optimize tasks on mobile devices is examined.

1) *Sensing optimization:* There are several approaches for lowering the energy consumption of mobile devices based on optimizing sensor usage. Adaptation of QoS parameters is one attempt that can be used to achieve this goal. In particular, the trade-off between service quality and consumed energy is a relevant matter in this context. Besides the adaptation of a service's performance, there is also the possibility of substituting sensors. In this case, specific sensors are substituted by other technologies which are capable of providing a comparable service while lowering the energy consumption [10][11]. If a complete substitution is not possible or needed, there is also the way of combining sensors, e.g., by triggering approaches [12]. Here, the acquisition of data from one sensor is triggered only when a second sensor reaches some kind of previously defined threshold. An example for this is the combination of a smartphone's accelerometer and its GPS sensor in the SenseLess concept introduced by Abdesslem et al. [13]. Their approach is to activate the GPS-Sensor only if the accelerometer detects the user being in motion. For evaluation, a user is equipped with two smartphones, one running the standard iOS positioning methods in a 10 second interval, the other one working with the SenseLess algorithm. The results showed, that SenseLess needed 85.5% less energy than the standard approach. The determined locations differed from 0.4m to 41m with an average value of 8m. But because

the energy consumption was determined on basis of the battery level, there is cause for inaccuracies through wrong interpreted up-and-down-movement, as well as due to the systems general energy consumption. As shown by Priyanta et al. [14], the combination of different sensors is to be used with caution and after individual evaluation. In their case, the computing of determined accelerometer data, which was gathered in a 10-minutes time interval with a frequency of 4Hz to 6Hz, consumed more energy than the location determination via GPS for 5 minutes, determining one location fix each minute. Schirmer et al. compared conventional GPS positioning approaches to custom triggering and polling concepts on iOS 4.3.1 and furthermore rated them concerning their energy consumption as well as their positioning accuracy [15]. With the polling approach, the authors tried to lower the energy consumption on a mobile device by the usage of a fixed sensor data request frequency. Thus, unnecessary data acquisition due to oversampling is about to be obviated. The triggering approach facilitated the usage of a device's built in GPS sensor in combination with its accelerometer. Therewith, the GPS usage was paused as long as no user movement was detected in order to save the device's resources during that time. For evaluation, Schirmer et al. recorded positioning fixes for a predefined route, including some predetermined movement breaks, with all of the three positioning concepts. The analysis of the result showed, that the performance of the polling was comparable to the standard iOS procedure. With the help of the triggering approach, the positioning procedure's energy consumption could be reduced, even though the positioning fixes' quality was significantly worse. Chon et al. introduce SmartDC, which not only aims at lowering the energy consumption while determining the users location, but also tries to predict the user's future positions and important places [16]. In order to accomplish that, the authors use unsupervised learning techniques, mobility prediction, and prediction of system usage based on Markov models. Using this approach, energy savings up to 81% and a prediction accuracy up to 80% have been reached. The maximum delay time was 160 seconds.

A complete substitution of sensors is used for Ambience-Fingerprinting [17], a technique that is especially suitable for indoor location determination, where no GPS signal can be received. Therefore, raw data of different kind is checked for specific attributes, so called fingerprints. An early concept of Pirantha et al. enables the user to discover his position via ultrasonic and radio fingerprints within a cm-accuracy [17]. Azizyan et al. present SurroundSense, a system for location determination on basis of environmental fingerprints like the spectral composition of noises or visual signatures [18]. The authors also analyze the cumulation of existing fingerprinting techniques, such as motion, noise, acceleration, brightness, color, and spectral contents of WLAN signals. In an evaluation with four users in predefined positioning clusters, 93% of all positions could be determined correctly. However, the consumed energy was not compared to the consumption of standard Software Development Kit (SDK) methods.

2) *Optimizing data transfer:* Due to the extensive energy consumption of data transfer and communication technologies, this field provides a lot of possibilities for achieving energy

savings. 2G, 3G, and 4G networks each use a different amount of energy for different tasks. Hence, already by choosing the best suited technology for each task, the overall energy consumption can be lowered. For example, 2G networks are suited better for calls than 3G networks, but 3G networks are much more efficient for data transfer [19]. Perucci et al. conclude that – even though additional power is consumed by the handover process needed for changing networks – significant energy savings can be achieved by consistently switching to a 2G connection for calls [20].

Furthermore, both Balasubramanian et al. [19] and Falaki et al. [21] reveal that not only the overall size of the transferred data is responsible for the consumed energy, but also the size of single transferred packages: Especially, small packages cause a transfer overhead up to 40%. Moreover, the Transmission Control Protocol (TCP) requires multiple transfers of the same packages due to package loss. To this end, Falaki et al. propose a bigger server side buffer and an optimization of the usage rules for the transfer networks in order to solve these problems, predicting energy savings up to 35%.

Another transfer-related optimization approach is TailEnd, provided by Balasubramanian et al. [19]. TailEnd divides applications into two groups, i.e., applications that tolerate delays in data transfer and those, which can profit from data prefetching. Based on this classification, data is loaded in bundles to minimize the device's staying in the high energy state of the IEEE 802.11 standard. Additionally, TailEnd makes usage of the advantages of the different transfer networks. In comparison to other approaches, 60% more newsfeeds and up to 50% more search results for web requests could be processed while consuming the same amount of energy.

3) *Using History and data-based prediction:* The analysis of user preferences and interaction patterns, as well as the usage of data mining techniques on historic or context data can reveal precious insights concerning the energetic regulation of a mobile system. For example, calendar or appointment specific data can be used to predict a user's current and future whereabouts. Predictions about the device's future energy level, combined with the information about upcoming tasks can be used to adapt the energy balance oriented to tasks rated critical by the user [22]. Oliver et al. succeeded in predicting the energy level of mobile devices correctly for a time slot of one hour with an error rate of only 7%. Within 24 hours the error rate was 28%. In order to achieve that, they classified the gathered data of more than 17,300 BlackBerry users and clustered it afterwards [23][24].

Trestian et al. conducted a network-based study which should reveal relations between the user, his movement patterns and used applications, in order to predict common interaction patterns. Their results indicate that the usage of specific applications is significantly related to the user's current movements or his location [25].

III. REQUIREMENTS

As shown in Section II-B, there are numerous approaches, which address single optimizations for specific system components or services. None of them tries to wrap all existing

concepts in one architecture, which organizes them in a modular way and provides requested services with an adequate output quality. In order to create a holistic energy management system which is capable of providing custom tailored services to a user, the following requirements for such an architecture can be identified:

- 1) **Universal validity and responsibility:** The energy management system is solely responsible for managing all resources of the system and the access to them. It receives all service and resource requests and answers them in an adequate and energy efficient way.
- 2) **Context and user awareness:** Typical context data like the current time and location, upcoming tasks, individual user preferences or social relations, as well as the systems current energetic state are used to provide services in a customized manner respecting the user's situation and whereabouts.
- 3) **Modularity:** A holistic energy management system is comprised of a multitude of different components, each responsible for different subtasks. In order to benefit from existing and future ideas in each of these areas, functionality is encapsulated in only loosely coupled modules, which can be altered or replaced without affecting the rest of the system.
- 4) **Scalability and extensibility:** New energy optimization concepts or services can get installed in an easy way, the user is able to use improved modules without non-trivial update processes.
- 5) **Definition of generic interfaces:** To make the integration of new modules possible, a definition of generic interfaces oriented on the lowest common denominator of installable modules is necessary.
- 6) **Adaptive data collection:** If data collection is required (e.g., sensor data), the system selects a service that acquires data adaptively to given preconditions. This may be the system's current energy state, the service quality as demanded by the service requester or forecasts concerning needed energy for future tasks. Furthermore, collected data needs to be cached for later usage.
- 7) **Task prioritization:** In order to take the user's preferences concerning critical tasks into account, the system is able to adhere to task priorities and assign more privileges to higher prioritized tasks.
- 8) **Clarity and comprehensibility:** The architecture must be structured in a clear and comprehensible way. This is essential to enable developers to integrate new optimization concepts or module packages into the existing system.

IV. EMMA - AN INTEGRATED MANAGEMENT APPROACH

Based on the requirements identified in Section III, we now present a holistic approach for managing resources in a user-centric and context-aware way, coined *EMMA (Energy Management Middleware Architecture)* [1]. Furthermore, EMMA allows an easy integration of existing and future energy optimization concepts. It provides its services in an adaptive,

context- and preferences-aware manner, controls service parameters and monitors the system's energy state. EMMA runs continuously in the background and is the exclusive interface to the system's resources (such as sensors) and services for other applications. Thus, EMMA is able to coordinate resource demands of all applications and to fulfill their requests in the most energy-efficient way.

A. Main components

EMMA consists of two main components, namely, the Control Unit and the Service Provider. These and their sub-components are depicted in Figure 1. In the following, the components are described.

1) *Control Unit:* The Control Unit is the central component dedicated to respond to service requests, select appropriate services based on context, preferences and demands, and monitor services and the system's energy state. If required, it adapts service parameters to ensure execution of critical tasks. In order to make intelligent decisions, the Control Unit employs information obtained from sub-components such as the Energy Manager, the Schedule Manager and the Service Identifier. Service requests, responses, and communication with the Service Provider, as well as among the subcomponents themselves are handled by a dedicated Controller. The Energy Manager analyzes the energy status of the whole system, calculates energy consumption of individual tasks and tries to predict future energy levels. These results can be combined with information about charging opportunities in vicinity or in near future, to reserve energy for future tasks or to preserve execution of critical tasks by performance adaptation for the longest time possible. The Service Identifier selects the most appropriate service for any given service request from the whole set of available services – as provided by the Service Provider – based on service parameters and quality of service characteristics. In general, service selection algorithms are geared towards a specific subset of services, since service parameters vary for different kinds of services. In order to monitor active and upcoming services, the Schedule Manager keeps track of currently running and future tasks, as well as respectively associated information.

2) *Service Provider:* The Service Provider is a passive component which does not contain any logic or executive power. Its core task is the integration and administration of service modules, as well as the provision of all requested services, the registration of needed callbacks and the caching of sensor data. To achieve that, the Registry Manager scans the folders containing all installed modules and corresponding manifest files, in order to identify added or removed modules for installation or removal. Afterwards, the current system state is saved in the Registry Cache, which provides these information for module identification matters to the Control Unit. As soon as a module was identified and released for usage by the Control Unit, the Service Provider creates a new service object and integrates the identified module in order to use its unique functionalities. The service is accessible via specific interface methods (see Section IV-D).

In order to avoid redundant acquisition of raw data, the Service Provider caches sensor data or other requested information

and provides it, if needed. In case that granularity, accuracy, and recency of cached data comply with the demanded QoS-parameters contained by an incoming service request, it can be answered by returning this data. The advantage of this procedure is obvious: By responding to requests with cached data, the instantiation of a new service object can be avoided and existing resources can be preserved for upcoming tasks.

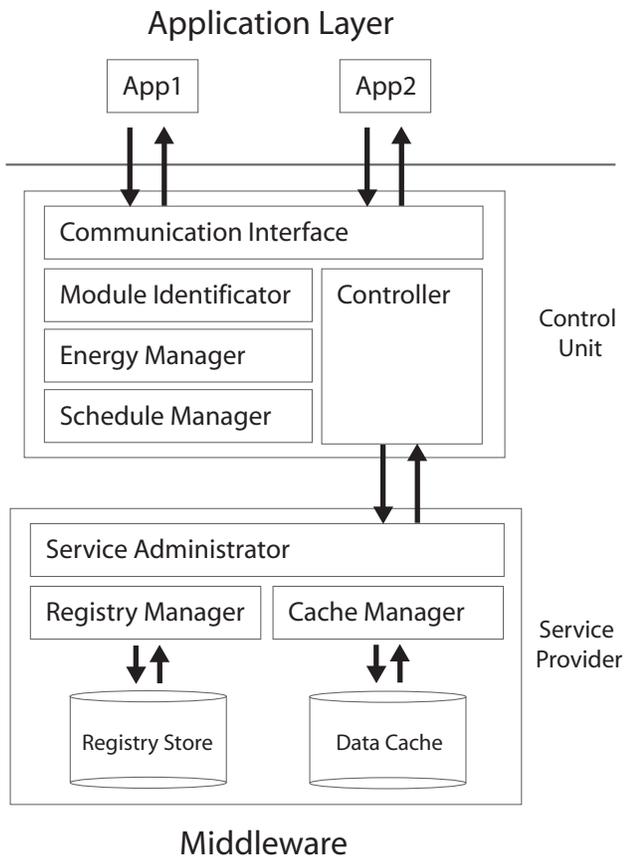


Figure 1. Overview of the EMMA concept and its components.

B. Modular service concept

A core feature of EMMA is its modular service concept. A service is defined as any functionality, that a user or an application might request, e.g., data transfer, acquisition of sensors data or media playback (see Section IV-C). In order to maximize the efficiency of providing these services, EMMA uses a highly modular concept. To this end, for each functionality a service class is defined in a formal way. Each service class might use different technologies to provide a requested service and the selection of a specific module might depend on diverse request properties, e.g., on the demanded service quality or the system's currently available energy. The internal logics of these different technologies are wrapped completely transparent in a service module and it is neither visible to requesting applications, nor the EMMA architecture

itself. The number of possible service modules per service class is unlimited, and the logic that is wrapped inside the module package is completely up to the module developer. The only thing, which needs to comply with EMMA's development guidelines, is the proper implementation of the predefined interfaces of the parent service class, as well as the delivery of an associated manifest file. The manifest contains information about the modules QoS parameters, as well as its parent service class and must be transcribed in a standardized way.

Later on, after the best suited module for a specific use case was identified (see Section IV-E), it is integrated into a service object at runtime (see Section V-B). Apart from the growing diversity of available technologies, this simple plug-and-play integration is another advantage of EMMA's modular concept. It enables optimized logic and improved technologies being integrated without the necessity for a complete system update (see Section V).

C. Requesting services

In order to answer an incoming service request in an adequate manner, several processes need to be executed. This is due to the necessity of considering the user's context and preferences, as well as the most recent energetic system state. Figure 2 illustrates the handling of a service request in context of a positioning service. As soon as a new service request was received at the communication interface, the parent class type of the requested service is extracted from the request's data package. This information is necessary to check if another instance of the requested service class is already active. If so, the service quality provided by the already active service object is checked and compared to the requested minimum QoS (1). E.g., in context of a positioning system, the minimum QoS-parameters would be matched if the delivered positioning accuracy is the same or higher than the requested. Furthermore, also the update frequency for new positioning fixes should be equal or higher. If a service, which covers the demanded quality requirements is active, it can be used for replying to the request by registering a new callback and return it in a response package (2). Thus, the request was answered and the procedure terminates successfully. But sharing of services is not possible for all kinds of requests, e.g., a data transfer service cannot be shared due to individual parameters such as its destination address or the data to transfer. In this case a new private service object needs to be created and returned.

If there is no comparable service running, the system checks if the demanded service is a single request or if it requires periodic updates (3). As long as no periodic updates are required, an inquiry for cached data is directed to the Service Provider. Cached data is usable if a) it is output of the requested service class, b) it is of the demanded recency, and c) it is corresponding with the QoS-parameters described in the requests data package. If there is cached data of this kind (4), it can be packed into a response package and returned to the requester (5). In this case, the procedure terminates successfully.

Else, if the request was not of a single nature, or if there is no data of sufficient quality in the cache, the system checks for the

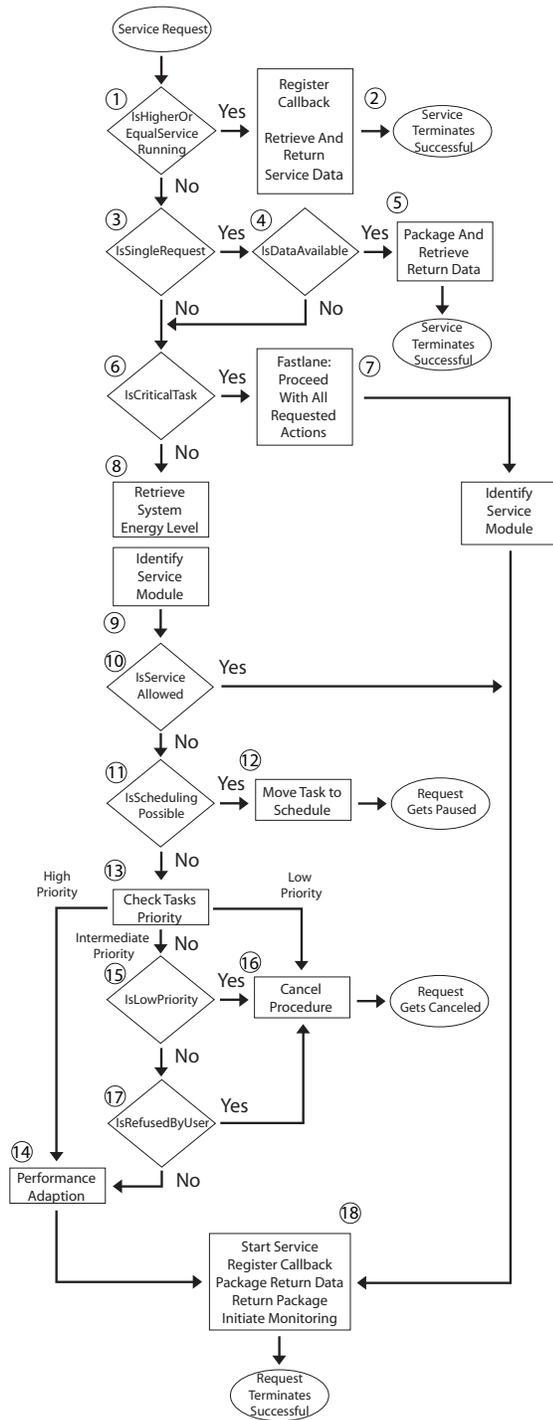


Figure 2. Handling of a service request with EMMA

request's priority in order to rate it critical or not (6) compared to other tasks. Critical tasks possess the highest priority in the EMMA system (see Section IV-G); their priority is rated critical by the user. Only special services, e.g., the device's emergency call function, are automatically classified as critical.

If a request is rated critical, all demanded procedures are executed without further examination and disregarding any relevant system states (7), e.g., the current battery level. The response is returned by undertaking all needed actions (18) and the procedure terminates successful.

Failing this, a system check regarding its current energetic state is carried out (see Section IV-F), in order to discover potential resources applicable for new tasks (8). In favor of this goal, the energy consumption of currently active services is considered as well as predicted consumption of services which are scheduled for future usage. Additionally, information about the current battery level or the user's position (e.g., nearby power supplies) is also respected. That is why even with a drained battery the energetic system status of a device could be rated as non-critical, because the user is able to easily charge his phone.

If it turned out in step (8), that there is enough energy for the new service, the identification of a suitable service module is initiated (see Section IV-E). For this purpose, the Control Unit's Controller obtains a list containing all available service modules of the requested service class and forwards it to the Module Identifier. This unit tries to identify the best suited module by processing the class specific identification algorithm in combination with the QoS-parameters provided by the requester (9). In case that there is no module belonging to the desired service class, the procedure is terminated. If there are modules of the desired service class, but none of them matches the requested QoS in an exact way, a pointer to the module with most similar features and performance parameters is returned. In a worst case scenario, no module can be identified. In an unfavorable scenario, a module with a higher energy consumption than allowed or with a lower service quality than demanded is identified. That is why the performance and consumption parameters of a module again are verified and compared to their counterparts given by the system. Purpose of this check is to determine if the module's features are compatible to the system's energy state (10), before a new service object is instantiated combined with the integration of the identified service module. If a service creation is not possible, the system tries to schedule the task (12) without the need to cancel it. This procedure provides the feature of resuming to a task as soon as new resources become available again. But due to individual peculiarities, this is not possible for all services. Reasons for that may be a high service urgency, because of the service's individual features or due to a user's preferences; other services may only be reasonable at a given time. E.g., a navigation service is likely to be needed immediately, while a transfer service could possibly get scheduled for a future point in time. The decision of scheduling tasks or not is carried out by processing a given flag in the requester's package whether a task is schedulable or not. If not, the system continues with the verification of the request's priority (13) (see Section IV-G). The priority flag is also included in the request's data package and was extracted from the users preferences as well as by analyzing the devices usage history. After its extraction, the priority can be compared to priorities of other tasks. If the new task possesses the highest priority (14), compared to the tasks that are responsible

for scheduled or currently active services, the system tries to free energy for the new task by throttling, canceling or pausing active services. But the adjustment of active services needs to be handled with care to avoid possible inefficiencies. E.g., canceling a transfer service during a data transmission process could force the new transfer of the whole amount of data on resuming to a task. This would lead to a higher energy consumption compared to just finishing an ongoing transmission. If an upcoming task possesses the lowest priority level, the procedure is canceled immediately and the request gets marked as a failure (16). In other respects, if the task is prioritized with an intermediate level, the system again tries to adjust, pause or cancel services owned by lower prioritized tasks. If that is not possible or the additional provided energy is not sufficient, the user is invited to choose which services are important to him and which are likely to be canceled. In order to keep up the system's usability, this approach is only used in combination with higher prioritized tasks. Additionally, all taken adjustment actions are logged to enable a later on analysis of the systems behavior. In case that a service was approved in (10) or (7), or if enough energy could be freed after the prioritization procedure, a new service object, containing the identified module, gets created, followed by the preparation of a return package (18).

Generally, return packages can contain three different types of data: 1) a callback registered with a running service instance (2), 2) requested low level data sets provided by the cache (5) or 3) a freshly created private service object. After service usage, the service object is released by the requester and is not available for usage any more.

D. Communication interfaces

The EMMA concept defines a fixed set of internal and external interfaces used for communication. Internal interfaces cover up all routing information and services, e.g., to realize a working connection between a service object and its integrated service module (see Section V). In contrast, external interfaces are responsible for processing service requests and responding to them. Therefore, data bundles are used to maintain the communication between EMMA and any applications requesting a service. These bundles contain the ID of the requested service class, as well as information about the requester's service priority, the expected service runtime and QoS-parameters.

The structure of corresponding response bundles is similar. If a request is answered positively, the reply contains the requested service, as well as information about its actual instantiation, as the returned service may either be a callback function the requester can use, a dedicated service object or simply the requested data. The information about its actual instantiation facilitates correct data unmarshalling.

E. Identification of suitable services

In order to be able to choose the most appropriate service module, EMMA executes an individual identification algorithm for each service class. This is necessary to ensure that the performance of the chosen module matches both, the energy

deallocated by the system and the demands of the service requester. Reasons for the necessity of having separate selection algorithms for each class are the different numbers of parameters, which are necessary for an adequate description of a service's energy consumption and its QoS-characteristics.

The simplest case of describing a module's characteristics is by the usage of two different performance parameters, which form a performance set. One parameter always represents the module's energy consumption, the other one describes a corresponding QoS-specific feature. Each module can possess any number of performance sets describing varying pairs of consumption and service quality related to different modes of operation. In this two dimensional space a target area can be identified between the point-of-origin and the intersection of the values of deallocated energy and requested service quality. In the following, the identification algorithm iterates across all existing modules of the requested service class and checks if any set of their performance parameters is located within the target area. If only one module can be found, this module will be selected. If several modules match the given thresholds, the one with the least energy consumption is selected. Otherwise, if no module can be found, the target area is extended and the same procedure is started again.

The complexity of the individual identification algorithms depends on the number of features, which are necessary for describing a module's performance. According to the current concept state, the mere provision of a service is regarded more important than the meeting of given performance parameters or the approved energy consumption. In case that no comparable module is offering the same service at a lower energy consumption, even a module with a much higher consumption than approved can be provided in order to meet the user's demands.

F. Continuous monitoring

In order to adapt the system's performance and to keep critical services running as long as possible, EMMA continuously monitors the system's energy state. Therefore, not only the current energy consumption but also the predicted consumption for future tasks, which can be determined by analyzing the active and upcoming tasks' schedules, combined with prediction techniques (see Section II-B3), is included. Additionally, relevant context information is considered, e.g., loading stations near a user's position (see Section IV-H). Each monitoring cycle is triggered by new incoming low level information concerning the system's power supply, e.g., the most recent battery level or changes in charging state. In order to visualize the monitoring process, Figure 3 shows each individual step which needs to be undertaken to control and adapt the system's resource usage. First, the system's current energetic state is determined by the Control Unit (1). The Schedule Manager provides information about effectively active services and the energy consumed by them. Together with predictions concerning the amount of energy that is needed for these and other upcoming tasks in the future, it can be forecasted if the energy will last for these duties (2). If there are no worries about the upcoming energy supply

and if there is no need - or not enough resources left - to upgrade the service quality of currently active tasks, the monitoring procedure terminates without further actions (5). Else, measures need to be initiated to adjust the system's energy balance (4). Generally, the system's performance can be adjusted into two directions: Given that either more energy than expected is available and the system's performance has been throttled before or a request demands a higher service quality than currently granted, the performance of a service as well as its QoS can be increased. In contrast, if there is less energy available than expected or a new high-priority service is started, the performance of active services can be throttled in order to free new resources. After undertaking demanded performance adaptations, the monitoring process jumps back to step (1) and the system's energetic state is checked again. If no further adaptations are necessary, the process terminates in (4).

To adapt the performance of a service module, it can be modified within its possible performance spectrum, e.g., throttle some or all of its individual performance attributes. If there are more adaptations needed, e.g., because of a better service quality or if even more energy is needed than the adjustment of a specific module can provide, the whole module can be replaced by another of the same service class but with different performance parameters. The adaptation of service performance commonly results in a change of service quality.

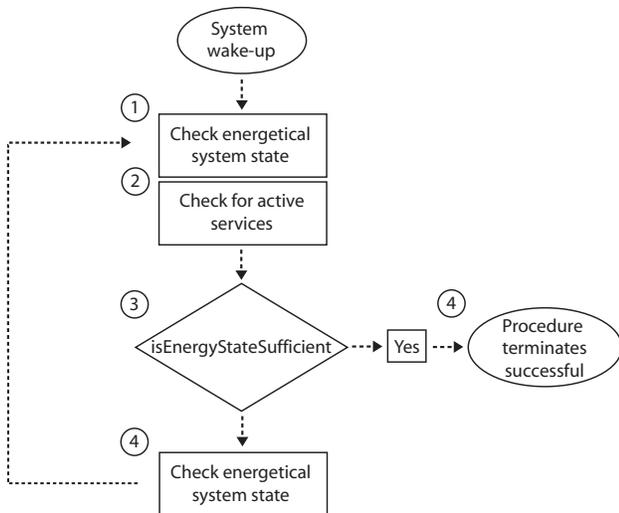


Figure 3. General overview of the EMMA concept and its components.

G. Task prioritization

Prioritizing tasks is one of EMMA's core concepts and not replacable for two different reasons. On the one hand, in order to make decisions concerning performance adaptations, a comparison as well as an evaluation of the scheduled service's performances is needed (see Section IV-F). On the other hand, there are services and applications with a critical priority,

which means their execution has to be enforced despite a lack of energy or other resources.

Furthermore, as long as no user specific preferences concerning a service's priority exist, each service gets assigned with the standard priority `PRIORITY_MEDIUM`. In order to take account of the user's personal preferences, EMMA provides a granular priority concept. The priority levels, featured by the prototypical implementation are `PRIORITY_CRITICAL`, `PRIORITY_HIGH` and `PRIORITY_LOW`. Afterwards, they can easily be extended with custom priority levels. The declaration of a service as critical is meant to ensure its execution, even under unfavorable circumstances.

Moreover, the usage of priority tags enables the constitution of hierarchical layers between all existing services, applications and tasks, which results in an order for sequential task processing. The processing direction itself is determined by the current use case. E.g., if new energy resources need to be freed, services from the lowest hierarchical layer are adjusted, paused or canceled in the first instance. Else, if there is an existing energy surplus that allows a performance lift of before throttled services, services in the highest hierarchical layer are considered first.

H. Energetical relevant context

The usage of context with energetic relevance is also one of EMMA's key features. It is especially needed for making decisions concerning the usage of the mobile device's resources. The Energy Manager component, which is part of the Control Unit, is responsible for that.

Concrete statements concerning the system's current energy state can be determined by appropriate Application Programming Interfaces (APIs), which are part of every recent smartphone operation system (e.g., the Android `BatteryManager` [26] or the iOS `UIDevice-object` [27]). In contrast, the prediction of forecasts concerning upcoming tasks and related resource drain is much more difficult. To achieve this, logs about application or service usage can be used as well as information about calendar events, the user's social environment or related preferences (see Section II-B3). If it is obvious that there are not enough resources available for future tasks, the QoS can be adjusted. Alternatively, if a context information indicates that at the time of an upcoming task a power supply is available, possible resource shortages could be ignored.

Anyways, predictions are not only used for making decisions towards resource management, they can also be used for periodical resource consumption monitoring. If it is obvious that enough resources for the lift of a service's performance quality are available, some services can be selected for a performance increase by using the before described prioritization system.

V. IMPLEMENTATION

In the following, some technical details concerning the general implementation of EMMA's prototype as well the construction of individual service objects are presented.

A. Basic parameters

In order to assess the feasibility of our approach, we developed a prototypical application of EMMA with the aid of the Google Android API (target version 18). In its current form, however, the prototype was not yet integrated as an exclusive service manager into the operating system. Instead, due to access and rights restrictions of the Android SDK it has been implemented as part of the application layer. Our goal in this first version was to evaluate the technical feasibility of EMMA's core concepts without analyzing any possible increase in energetic efficiency in detail, yet.

In order to cope with EMMA's claim for modularity, the logic of service modules is capsuled in `.dex`-archive files, whose content is described both machine- and human-readable in corresponding XML manifest documents. The latter contain all information about a module's performance, nature, and energy consumption and can hence be used to dynamically identify any modules matching a service request. Our prototype is able to respond to incoming requests for either a continuous or a one-time positioning service by selecting the most appropriate service module, taking into account the current energy state of the device, as well as context information and currently active, or scheduled services.

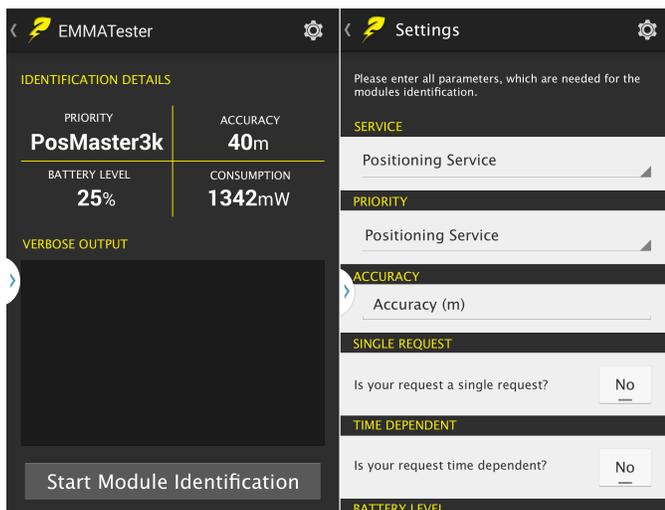


Figure 4. Screens of the EMMA prototype application.

To achieve this, the device's current energy consumption is determined at first. In order to keep the implementation overhead for the energy measurements as low as possible, the energy consumption is approximated by a simple model based on test results obtained from a HTC Desire using the PowerTutor application [8]. By combining these information with knowledge about active and scheduled services as well as their estimated runtime, it is now possible to calculate the energy remaining for new services. In order to identify a suitable service module, the requested QoS parameter contained in a request bundle and the amount of remaining energy are considered for building a target area (see Section IV-E). After identification, the chosen module is integrated into a singleton

service object. According to the current concept, the identified module is not restricted in its energy consumption once it was installed, but it can be replaced by a more economical one if its consumption is extensive.

In order to be able to use the module's functionalities after its integration, a standard callback object provided by the Android SDK is installed in the service object, serialized and returned to the service requester. The latter is now able to listen for positioning updates after rebuilding the callback from its serialized version.

B. Constructing service objects

From a technical point of view, a service object consists of several individual components, which are visualized in Figure 5. In case of our prototypical implementation, each individual service object is derived from a basic service object and expands it with individual functionalities. Generally, a service object can be seen as an empty hull not realizing any kind of own logic. It merely implements all predetermined integration and communication gateways, in order to ensure a smooth integration of external logic modules. For the most a basic service object encapsulates all available information related to the service class it represents as well as the service modules which are allowed to become integrated. Moreover, it provides a standardized interface and related functions for accessing the service logic inside of the integrated module. On the upper left, all registered callbacks, which are organized in a list are shown. If a service is suitable for simultaneous usage by multiple users, a new user callback is registered here. This approach prevents the system from an unnecessary, parallel execution of multiple instances of the same service. Additionally, each registered callback contains a unique ID, which is returned to a service requester after its registration. The ID ensures, that a callback can be identified and removed after service usage.

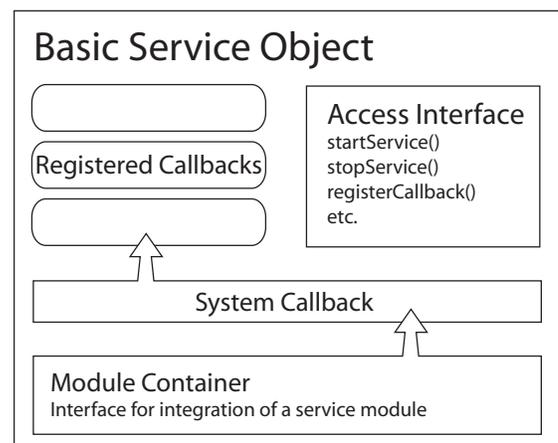


Figure 5. Schematic representation of a service object.

Furthermore, each service object possesses an individual and private system callback, which manages the internal communication to the service module. It enables the reception of

messages directly from the service module and mirrors them in a second step to each registered user callback.

On the upper right side of Figure 5, examples for possible access procedures are shown. These are necessary for a direct service or logic access, e.g., for delivering data sets or for starting or stopping a service, etc. All required methods for accessing a service class are defined in a unique way, which needs to be implemented by each module of this specific service class. Additionally, the module's container implements some further gateways for integrating the service module into its service object. At runtime, all logic contained in the module archive is loaded dynamically into the program code by using the Android-specific, reflection-based DexFile [28] and DexClassLoader-classes [29].

VI. CONCLUSION AND FUTURE WORK

By introducing EMMA, we provided a holistic, modular concept for individual and context sensitive energy management, which meets the requirements stated in Section III. The modular concept applies facilities for user-centric service provision and renders the usage of partial optimization concepts as well as of their energetic savings potential. Furthermore, the provided architecture makes the integration of third-party logic a more trivial task for users and developers.

EMMA in its current state is to be considered a basic architectural framework. Lots of details need an initial clarification or further composition, e.g., the specification of different service classes and service modules. The process of identifying and providing services needs further improvements, as well. In its current state, tasks need a manual prioritization undertaken by the user. In order to automate this process, the usage of training or machine learning algorithms is conceivable. Furthermore, a capable security concept must be provided due to the possibility of including third-party code dynamically into the system at runtime. This feature indicates that additional security measures are inevitable in order to avoid the execution of malware within EMMA and to avoid the system's abuse.

In context of a following expert's discussion consisting of 8 participants with IT background, which was aimed to evaluate EMMA's feasibility and utility, we found that the procedure of integrating modules and the conventions for developing them, as well as the interface descriptions for requesting services need to become more facilitated. In their current state, they seemed to be not as clear as needed for an unrestricted and easy system usage. In a final step, a second implementation of the EMMA concept with the goal of evaluating its actual energy savings potential is to be carried out. Especially, the integration of the architecture as part of the operational system proves to be a challenge, since current platforms like iOS or Android do not allow this without significant interventions in their system's core.

REFERENCES

- [1] A. Ebert, F. Dorfmeister, M. Maier, and C. Linnhoff-Popien, "Emma: A context-aware middleware for energy management on mobile devices," in *CENTRIC 2014, The Seventh International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*, pp. 48–53, IARIA, 2014.
- [2] A. K. Dey, *Providing architectural support for building context-aware applications*. PhD thesis, Georgia Institute of Technology, 2000.
- [3] P. A. Zandbergen, "Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning," *Transactions in GIS*, vol. 13, no. s1, pp. 5–25, 2009.
- [4] B. Li, J. Salter, A. G. Dempster, and C. Rizos, "Indoor positioning techniques based on wireless lan," in *LAN, First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, Citeseer, 2006.
- [5] M. Schirmer and H. Höpfner, "Senst*: approaches for reducing the energy consumption of smartphone-based context recognition," in *Modeling and Using Context*, pp. 250–263, Springer, 2011.
- [6] J. Manweiler and R. Roy Choudhury, "Avoiding the rush hours: Wifi energy management via traffic isolation," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pp. 253–266, ACM, 2011.
- [7] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app?: fine grained energy accounting on smartphones with eprof," in *Proceedings of the 7th ACM european conference on Computer Systems*, pp. 29–42, ACM, 2012.
- [8] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pp. 105–114, ACM, 2010.
- [9] R. Mittal, A. Kansal, and R. Chandra, "Empowering developers to estimate app energy consumption," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 317–328, ACM, 2012.
- [10] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 299–314, ACM, 2010.
- [11] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 315–330, ACM, 2010.
- [12] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox, "Enloc: Energy-efficient localization for mobile phones," in *INFOCOM 2009, IEEE*, pp. 2716–2720, IEEE, 2009.
- [13] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: energy-efficient mobile sensing with senseless," in *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*, pp. 61–62, ACM, 2009.
- [14] B. Priyantha, D. Lymberopoulos, and J. Liu, "Littlerock: Enabling energy-efficient continuous sensing on mobile phones," *Pervasive Computing, IEEE*, vol. 10, no. 2, pp. 12–15, 2011.
- [15] M. Schirmer and H. Höpfner, "Senst*: approaches for reducing the energy consumption of smartphone-based context recognition," in *Modeling and Using Context*, pp. 250–263, Springer, 2011.
- [16] Y. Chon, E. Talipov, H. Shin, and H. Cha, "Mobility prediction-based smartphone energy optimization for everyday location monitoring," in *Proceedings of the 9th ACM conference on embedded networked sensor systems*, pp. 82–95, ACM, 2011.
- [17] N. B. Priyantha, *The cricket indoor location system*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [18] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, pp. 261–272, ACM, 2009.
- [19] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy consumption in mobile phones: a measurement study and implications for network applications," in *Proceedings of the 9th ACM*

- SIGCOMM conference on Internet measurement conference*, pp. 280–293, ACM, 2009.
- [20] G. P. Perrucci, F. H. Fitzek, G. Sasso, W. Kellerer, and J. Widmer, “On the impact of 2g and 3g network usage for mobile phones’ battery life,” in *Wireless Conference, 2009. EW 2009. European*, pp. 255–259, IEEE, 2009.
- [21] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, “A first look at traffic on smartphones,” in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 281–287, ACM, 2010.
- [22] N. Ravi, J. Scott, L. Han, and L. Iftode, “Context-aware battery management for mobile phones,” in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pp. 224–233, IEEE, 2008.
- [23] E. Oliver and S. Keshav, “Data driven smartphone energy level prediction,” *University of Waterloo Technical Report*, 2010.
- [24] E. Oliver, “Diversity in smartphone energy consumption,” in *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students*, pp. 25–28, ACM, 2010.
- [25] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, “Primary user behavior in cellular networks and implications for dynamic spectrum access,” *Communications Magazine, IEEE*, vol. 47, no. 3, pp. 88–95, 2009.
- [26] “Android Developers - Monitoring the Battery Level and Charging State.” <http://developer.android.com/training/monitoring-device-state/battery-monitoring.html>, 2015. [Online; accessed 06-January-2015].
- [27] “Apple Developers - UIDevice Object.” https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIDevice_Class/index.html, 2015. [Online; accessed 06-January-2015].
- [28] “DexFile - Android Developers.” <http://developer.android.com/reference/dalvik/system/DexFile.html>, 2014. [Online; accessed 25-July-2014].
- [29] “DexClassLoader - Android Developers.” <http://developer.android.com/reference/dalvik/system/DexClassLoader.html>, 2014. [Online; accessed 25-July-2014].

Server and Path Selection in a Light Architecture Content Streaming System with Dual Adaptation

Eugen Borcoci, Marius Vochin, Mihai Constantinescu,
University POLITEHNICA of Bucharest
Bucharest, Romania
emails: eugen.borcoci@elcom.pub.ro,
marius.vochin@elcom.pub.ro,
mihai.constantinescu@elcom.pub.ro

Jordi Mongay Batalla, Warsaw University of Technology
/ National Institute of Telecommunications
Warsaw, Poland
jordim@interfree.it
Daniel Negru, LaBRI Lab, University of Bordeaux,
Bordeaux, France
daniel.negru@labri.fr

Abstract — Media content streaming and delivery is nowadays a high popularity service in Internet. Complex architectures like Content Delivery Networks, Content Aware/Oriented Networks have been proposed, where information/content objects are treated as first-class abstractions. As alternative, light architectures, are investigated and implemented, working on top of the current IP networking technologies. In both types of architectures, appropriate content server and path selection constitute the primary set of actions to be performed in the content delivery systems. Such a problem belongs to the general class of multi-criteria optimization problem, having (in our case) as input, some information on servers, network and user context. This paper contains an extension of a preliminary work, focused on algorithms and policies for optimized paths and server selection. Simulation study results are presented here, to illustrate the better performance of multi-criteria optimization algorithms versus random path and/or server selection. Flexibility of the solution is emphasized, including the possibility to naturally add new criteria (business, policies) in the selection process. This work is a part of a larger effort, aiming to finally implement a subsystem in the framework of a content delivery light architecture system.

Keywords — Content delivery; Server selection; Path selection; Content-Aware Networking; Multi-criteria decision algorithms; Dual adaptation; Future Internet.

I. INTRODUCTION

The content orientation is an important trend recognized in the current and Future Internet [1][2]. Consequently, several solutions have been recently proposed, studied and implemented, aiming to better support the content oriented services. The *Information/Content-Centric Networking* (ICN/CCN) approach [3][4] revisits some main concepts of the architectural TCP/IP stack; the novelty is that in ICN/CCN the information/content objects are treated as first-class abstractions. In such architectures the intelligence and complexity of the network nodes are higher; the routers can perform content-based route computation (routing) and forwarding, caching and other content-oriented processing, leading to systems better adapted to the content requests and delivery, in comparison with traditional network. However, the cost of such systems is rather high, both due to the architectural changes and also due to the much higher

processing performance required from routers. Therefore, some more “light ICN” evolutionary solutions, preserves the main TCP/IP concepts, but introduce a degree of *Content-Awareness at Network* layer (CAN) [5]. Seen partially as an orthogonal solution, *Content Delivery Networks* (CDNs) improve the content services [6] by distributing the content replica to cache servers, located close to groups of users. However, all the above solutions involve complex management and control architectures, high CAPEX and significant modifications to be introduced by Service/Content Providers and Network Providers/Operators.

As alternative, over-the-top (OTT) solutions are investigated, where the high level services are delivered over the connectivity offered in current Internet. Here, a Service Provider (SP) is not (fully) responsible for the transmission of the information flows to the end-user; users access is done via the “public Internet”. An OTT-type SP could be an entity separate from the traditional Internet Service Provider (ISP). Also, combined solutions exist, with OTT Service Providers using the CDN Providers infrastructure to improve the quality of delivery.

A light architecture (OTT-like), for content streaming systems over the current Internet is proposed by the European *DISEDAN* Chist-Era project [7][8], (*service and user-based DIstributed SElection of content streaming source and Dual Adaptation*, 2014-2015). The business actors involved are: *Service Provider (SP)*, which is an entity/actor that delivers the content services to the users and possibly owns and manages the transportation network); *End Users (EU)*, which consumes the content; a *Content Provider* could exist, which is the owner of some *Content Servers*. However, *DISEDAN* does not deal with contractual relationships between the CP and SP; one may therefore assume, in a simplified model, that servers are also owned by the SP.

A novel concept is introduced based on:

(1) a *two-step server selection mechanism* (at SP and at EU) using algorithms that consider context- and content-awareness. An effective solution is constructed for the multi-criteria hard problem of best content source (server) selection, considering user context, servers’ availability and requested content.

(2) a *dual adaptation mechanism* consisting of Media adaptation (also called *media flow adaptation*) and content source adaptation (by *switching the streaming server*), when the quality observed by the user suffers degradation during the media session.

The proposed solution could be rapidly deployed in the market since it does not require complex architecture like CON/ICN, full-CAN or CDN.

This paper is an extension of the work ([1], ICSNC 2014 conference paper) on paths and server combined selection algorithms and policies applicable by SPs in a system having light content delivery architecture. This contribution additionally presents and analyzes a lot of experimental results. The acquisition of the input information for the selection procedure is out of scope of this work; it is supposed that such information is provided statically or dynamically (by measurements performed by a monitoring subsystem) and made available for the algorithm.

Section II is a short overview of related work, focused on multi-criteria optimization algorithms and their adaptation to our context. Section III describes at high level the overall system and outlines the server-path selection problem. Sections IV and V contain the main paper contributions, focused on paths and content server selection combined algorithm, simulation models and results. Section VI proposes modifications of the MCDA algorithms to allow introduction of SP policies, aiming to increase the system flexibility. Section VII contains conclusions and future work outline.

II. RELATED WORK

A. Multi-criteria decision Algorithms

This section is a short overview on some previous work related to *path-server selection* in content delivery systems, based on *Multi-Criteria Decision Algorithms (MCDA)*. The problem belongs to the more general one, known as *multi-objective optimization*. This has been extensively studied in various and large contexts of economics and engineering. The paper will not detail this. Few references are given at the end of the paper [9][10].

The general problem of multi-objective optimization is to minimize $\{f_1(x), f_2(x), \dots, f_m(x)\}$, where $x \in S$ (set of feasible solutions), $S \subset \mathbb{R}^n$.

A *decision vector* is defined as $x = (x_1, x_2, \dots, x_n)^T$.

One might have ($m \geq 2$) possibly conflicting *objective functions* $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m$ and we would want to minimize them simultaneously.

A set of *Objective vectors* is defined as images of decision vectors, where *objective (function) values* are :

$$z = f(x) = (f_1(x), f_2(x), \dots, f_m(x))^T$$

It is called a *feasible objective region* $W = f(S)$ the image of S in the objective space.

Objective vectors are said to be *optimal* if none of their components can be improved without deterioration to at least one of the other components.

A *decision vector* $x_- \in S$ is defined as *Pareto optimal* if there does not exist another $x \in S$ such that $f_i(x) \leq f_i(x_-)$ for all $i = 1, \dots, k$ and $f_j(x) < f_j(x_-)$ for at least one index j .

Figure 1 shows an example for Pareto front, where $x = (\text{server}, \text{path})$; $n = 2$, $x \in \mathbb{Z}^2$ (the paths and servers are identified through some positive integer indexes). We have $f(x) = (f_1, f_2) = (\text{srv_load}, 1/\text{path_avail_bandwidth})$, $m = 2$.

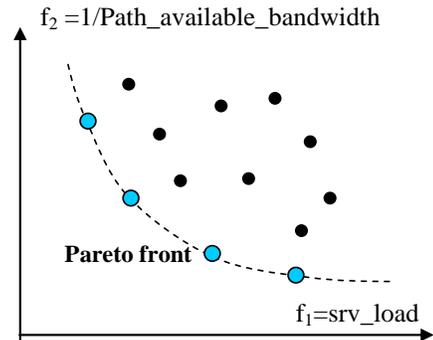


Figure 1. An example of objective space and Pareto front

Such optimization problems are in general NP complete, so, different simplified heuristics have been searched. A simple scalar approach maps the k -dimensional vector onto a single scalar value w by using an appropriate cost function $c()$, thus reducing the problem to a single-criterion one. However, information about individual components is lost. In the server-path selection problem several decision parameters are important, such as: server load and proximity, transport path (length, bandwidth, loss, and jitter).

Solutions have been searched, treating the decision variables separately and considering them as independent. Note that in our case this is only partially true, e.g., delay and jitter are clearly not independent variables. Therefore, modifications should be added to the basic algorithm to capture such effects and this paper proposes a solution.

In the following, we will use a simplified notation:

- identify the solutions directly by their images in the objectives space \mathbb{R}^m ; in other words, we define as decision parameters/variable the set v_i , $i = 1, \dots, m$, with $\forall i, v_i \geq 0$, where they are actually the values of the objective functions;
- S is number of candidate solutions; they are indexed by $s = 1, 2, \dots, S$;
- the image of a candidate solution s is $Sl_s = (v_{s1}, v_{s2}, \dots, v_{sm})$ represented by a point in \mathbb{R}^m .

Value ranges of decision variables may be bounded by given constraints. The selection process means to select a solution satisfying a given objective function conforming a particular metric.

One approach to solve the optimization problem is *reference level decision algorithms* [11-13]. This will be used in the paper. Considering the above notations, the basic algorithm defines two reference parameters:

- r_i = reservation level = the upper limit for a decision variable which should not be crossed by the selected solution;

- a_i =aspiration level=the lower bound for a decision variable, beyond which the solutions are seen as similar.

Without loss of generality one may apply the definitions of [14][15], where for each decision variable v_i there are defined r_i and a_i , by computing among all solutions $s = 1, 2, \dots, S$:

$$\begin{aligned} r_i &= \max [v_{is}], s = 1, 2, \dots, S \\ a_i &= \min [v_{is}], s = 1, 2, \dots, S \end{aligned} \quad (1)$$

Figure 2 shows a simple example, where $m = 2$.

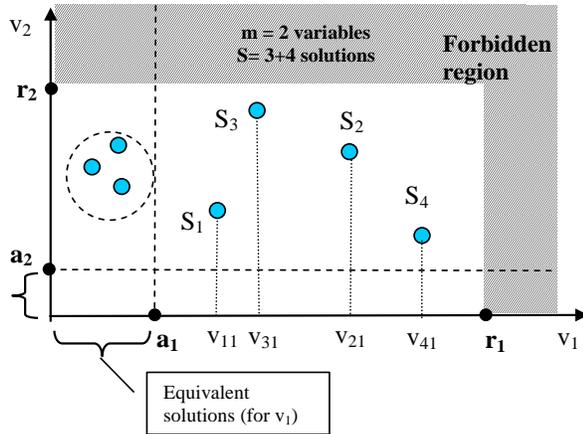


Figure 2. An example of reference level limits (aspiration and reservation)

In [13], modifications of the decision variables are proposed: *replace each variable with distance from it to the reservation level: $v_i \rightarrow r_i - v_i$* (so, increasing v_i will decrease the distance); normalization is also introduced to get *non-dimensional values, which can be numerically compared*. For each variable v_{si} , a ratio is computed, for each solution s , and each variable i :

$$v_{si}' = (r_i - v_{si}) / (r_i - a_i) \quad (2)$$

The factor $1/(r_i - a_i)$ - plays also the role of a weight. The variable having high dispersion of values ($max - min$) will have lower weights, and therefore greater chances to determine the minimum in the next relation (3). In other words, less preference is given to those variables having close values (reason: a given candidate selection among them will not influence significantly the overall optimum).

The algorithm steps are:

Step 0. Compute the matrix $M\{v_{si}'\}$, $s=1 \dots S$, $i=1 \dots m$

Step 1. Compute for each candidate solution s , the minimum among all its normalized variables v_{si}' :

$$\min_s = \min\{v_{si}'\}; i=1 \dots m \quad (3)$$

Step 2. Make selection among solutions by computing:

$$v_{opt} = \max\{\min_s\}, s=1, \dots, S \quad (4)$$

This v_{opt} is the optimum solution, i.e. it selects the best value among those produced by the Step 1.

The reference level algorithm has been used in several studies.

The work [15] proposes a decision process for network-aware applications, based on reference level MCDA with several metrics. The improvement (compared to the basic algorithm) consists in considering not only the currently selected server status, but also the system future state after the selection. The simulation results showed a slight gain versus the basic algorithm, while using the same information from the network level (server and link load).

The work [16] proposes and evaluates a multi-criteria decision algorithm for efficient content delivery applicable to CDN and/or ICN. It computes the *best available source and path* based on information on content transfer requirements, servers and users' location, servers load and available paths. It runs processes at two levels: 1. *offline* discovering multiple paths, and gathering their transfer characteristics; 2. for each content (online) request, finding the best combined server - path (reference level model). The following "use cases" are evaluated: *random server and random path*, combined with shortest single path routing protocol (current Internet solution); *closest server and random path*, (similar to the current CDN); *least loaded server and random path*; *best server and the path with more available bandwidth* in the bottleneck link. Simulation, using Internet large scale network model, confirmed the effectiveness gain of a content network architectures (i.e., having a degree of network awareness) and efficiency of the combined path-server selection.

The work [17] models and analyzes a simple paradigm for *client-side server selection*. Each user independently measures the performance of a set of candidate servers, randomly chooses two or more candidate and selects the server providing the best hit-rate. The algorithm converges quickly to an optimal state where all users receive the best hit-rate (respectively, bit rate), with high probability. It is also shown that if each user chooses just one random server instead of two, some users receive a hit-rate (respectively, bit rate) that tends to zero. Simulations have evaluated the performance with varying choices of parameters, system load, and content popularity.

The contributions of this paper w.r.t. previous work mentioned are summarized as: two-phase flexible selection procedure based on MCDA reference level algorithm, applicable with slight modifications for nine use cases (see Section IV); additional policy supporting modifications proposed for the basic algorithm, in order to capture different Service Provider strategies.

Note that other algorithms can be used to optimize the selection belonging to a different class like *Evolutionary Multi-objective Optimization Algorithms (EMO)* [18]. However, this approach is not in the scope of this study.

B. Dual Adaptation

The adaptation techniques of interest for DISEDAN have been: in-session media flow adaptation methods (*Dynamic Adaptive Streaming over HTTP (DASH)* [19-20]) and complemented by the *Content Servers (CS)* switching (this is partially similar to the first CS selection).

The DISEDAN DASH subsystem details are not in the scope of this paper. However, for the sake of completeness we gave a short description of it. This technology has been selected in DISEDAN for in-session media adaptation.

DASH was recently adopted as multimedia streaming standard, to deliver high quality multimedia content over the Internet, by using conventional HTTP Web servers [19-20]. It minimizes server processing power and is video codec agnostic; it enables automatic switching of quality levels according to network conditions, user requirements, and expectations. The DASH offers important advantages (over traditional push-based streaming) [21]. A DASH client continuously selects the highest possible video representation quality that ensures smooth play-out, in the current downloading conditions. This selection is performed on-the-fly, during video play-out, from a pre-defined discrete set of available video rates and with a pre-defined granularity (according to video segmentation).

III. DISEDAN SYSTEM SUMMARY

A. System Concept

The DISEDAN project proposes an *evolutionary and light architecture* for content delivery via Internet, multi-domain compatible. It works in Over the Top (OTT) style, involving more simple management and control in comparison to ICN/CCN. DISEDAN defines a novel concept having as main features, as mentioned in Introduction section: a. *two-step server selection mechanism* (at Service Provider (SP) and at End User) by using algorithms that consider context- and content-awareness; b. *dual adaptation mechanism during the sessions*, - which consists in media flow adaptation (based on segmented video content delivery by using *Dynamic Adaptive Streaming over HTTP (DASH)* [19-20]) and/or content servers switching (handover).

Note that DASH details and design are out of scope of this paper.

Figure 1 illustrates the system concept. The main business entities/ actors are those mentioned above: SP, EU, CS. The SP and possible Content Provider (CP) entities are not seen as distinct. Also, the full CS management is out of scope of this system. The connectivity between CSs and EU Terminals (EUT) are assured by traditional *Internet Services Providers (ISP) / Network Providers (NP)* - operators. The ISP/NPs do not enter explicitly neither in the business relationships set considered by DISEDAN, nor in the management architecture.

It is supposed that SP has knowledge on: Content servers identities, their status - including their availability in terms of content objects contained and their current available capacity to serve new requests. Also, the SP may (optionally) have some information about static and/or dynamic connectivity resources (at overlay level) in the network, between different servers and potential groups of users (the latter could be placed everywhere in the network). Such information - if it exists - comes from external entities (e.g, network operators) or even from the CSs, if they are capable to evaluate the characteristics of some paths.

A simplified scenario is described below (see Figure 3).

The End User issues a *Media description request* (action (1) in Figure 3) to SP. The SP analyses the requests; it evaluates the CSs status and after applying an optimization algorithms, returns to the user a *Media Program Description (MPD) file* - containing, among others, the identity of a selected server (or a ordered list of servers). The End User performs the final selection based on SP-delivered information and - possibly - based also on some local measurements. Again, an optimization algorithm could be used in this final selection. The End User selects Content Source 1 and starts to ask (HTTP requests) video segments, (2).

While receiving the segments, local quality measurements (3) are performed at End User Terminal (EUT) in order to guide the adaptation process. If, for some reasons, the received quality was poor, then the EUT may decide:

- either to apply DASH adaptation, i.e., to determine the rate for the next segment request and maybe ask the next segment with a lower bit rate (5a) from the same Content source, or
- to switch to another CS. In this case, it may initiate probing of other candidates (4) and finally switch the Content source (in our case Content Source 2 is selected as a new source).

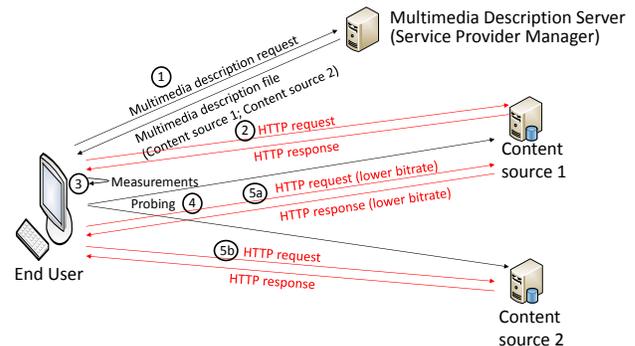


Figure 3. DISEDAN concept

B. System Architecture

Figure 4 shows a simplified high level view of the general architecture.

The Service Provider includes in its Control Plane:

- **MPD File generator** – dynamically generates Media Presentation Description (MPD) XML file, containing media segments information (video resolution, bit rates, etc.), ranked list of recommended CSs and, optionally - current CSs state information and network state (if applicable).
- **Selection algorithm** – runs Step 1 of server selection process. It exploits *Multi-Criteria Decision Algorithms (MCDA)* [9][15][16], modified to be applied to DISEDAN context, or *Evolutionary Multi-objective Optimization algorithm (EMO)* [18], etc.,

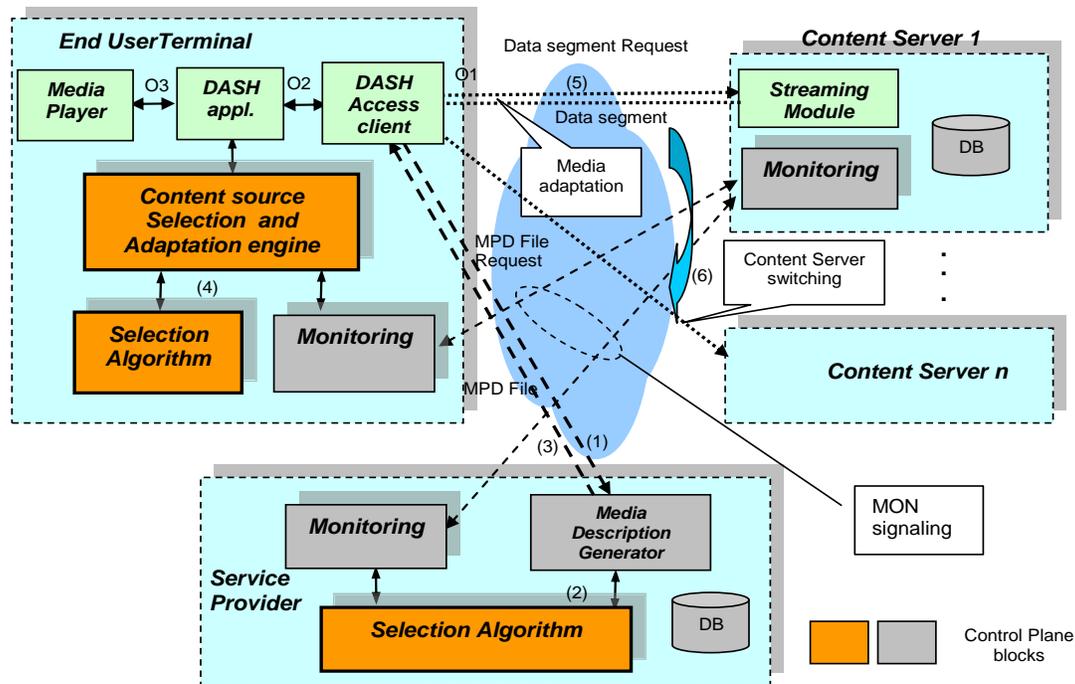


Figure 4. DISEDAN general architecture; DASH - Dynamic Adaptive Streaming over HTTP; MD – Media Description; DB – Data Base ; O1, O2, O3 – DASH Observation Points [ISO/IEC 23009-1]

to rank recommended CSs and media representations, aiming to optimize servers load as well as to maximize system utilization.

- *Monitoring module* – collects monitoring information from CSs and performs the processing required to estimate the current state of each CS.

The End User Terminal entity includes the modules:

- *Data Plane: DASH (access and application)* – parses the MD file received from SP and handles the download of media segments from CS; *Media Player* – playbacks the downloaded media segments.
- *Control Plane: Content Source Selection and Adaptation engine* – implements the dual adaptation mechanism; *Selection algorithm* – performs the Step 2 of server selection process. It can also exploit MCDA, EMO, or other algorithms to select the best CS from the set of candidates recommended by SP; It runs adaptation process; *Monitoring module* – monitors changing (local) network and server conditions.

The Content Server entity includes the modules:

- *Data Plane: Streaming module* – sends media segments requested by End Users;
- *Control Plane: Monitoring module* – monitors CS performance metrics (CPU utilization, network interfaces utilization, etc.).

The following functional steps are performed:

(1) EUT issues to SP a media file request.

(2) SP analyzes the status of the CSs and runs the selection algorithm (optionally the SP could make first, a current probing of the CSs); For each user request the SP could consider also the user profile, the policies of the SP for this user's class and other information at the SP side (e.g., states of the servers and possibly network-related information).

(3) SP returns to EUT a ordered list of candidates CS (SP proposal) embedded in a MD- xml) file.

(4) The EUT performs the final CS selection, by running its own selection algorithm.

(5) EUT starts asking segments from the selected CS. During media session, the EUT makes quality and context measurements. Continuous media flow adaptation is applied using DASH technology if necessary or (6) CS switching is decided. From the EU point of view the steps 1-2-3 composed the so-called Phase 1 and steps 4-5-6 the Phase 2.

During the receipt of consecutive chunks, the user's application can automatically change the rate of the content stream (internal DASH actions, - which are out of scope in this paper) and/or also can switch to another CS.

IV. PATH AND SERVER SELECTION OPTIMIZATION ALGORITHM

A two phase selection process is adopted here, similar to [9][15]. The Phase 1 is executed offline and computes candidate paths from servers to users. The Phase 2 applies a MCDA (reference level variant) algorithm and computes the best path-server solution, based on multi-criteria and also policies guidelines. Note that the multi-criteria algorithm is

flexible: any number of decision variables can be used, depending on their availability. For instance, in a multi-domain network environment it is possible that SP has not relevant or complete knowledge about end to end (E2E) transport paths. In such cases the list of available decision variables can be as well used. Another additional contribution here consists in modifying the reference algorithm, to include different SP policies concerning the importance of some decision variables with respect to others.

A. Network Environment

The content delivery might involve several network domains independently managed [4][5]. In a combined optimization procedure for path and server selection it is not realistic, from the real systems management point of view, to consider all details of the paths from the content servers to the users. Therefore (supposed in this paper and also in DISEDAN), the SP network awareness is limited, e.g., to knowledge about the inter-domain context, i.e., the inter-domain graph (where each network domain is abstracted as a node) and inter-domain link capacities, while considering the multi-tier organized Internet. The location (domains) of the potential groups of users and server clusters are also supposed to be known.

Figure 5 shows a generic example of a tiered structure network, containing several domains D11, ...D33 interconnected via inter-domain links. At the edges of this structure, groups of servers and users are connected to Tier 3 domains. In Figure 5, two possible paths from D33 to D32 are shown. The Phase 1 procedure will compute such similar paths between two edge domains.

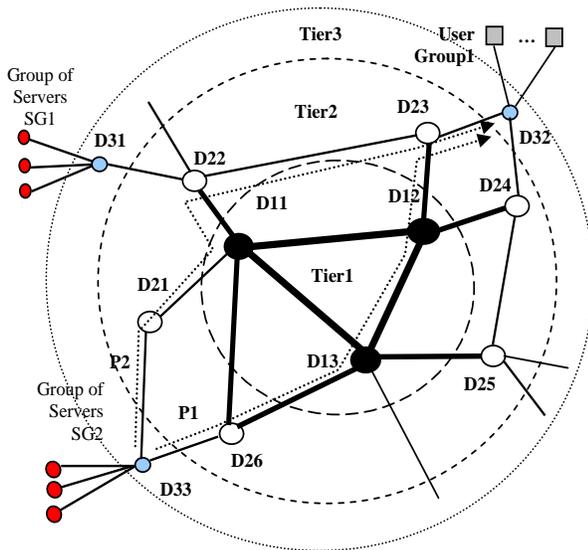


Figure 5. Example of a sample tiered network. P1 and P2 – paths from domain D33 to D32.

B. Use cases for path-server selection procedure based on MCDA algorithm

Several Use Cases can be defined for a combined algorithm, by considering several criteria for the path and server selection. Different metrics can be defined for paths and servers status evaluation. The path metric can be the simplest - number of hops, or a more powerful one (enabling better QoS assurance), e.g.: link cost=1/B, where B could be the static link capacity or the available bandwidth (dynamically measured). Also, constrained routing policies can be applied (e.g., related to bandwidth, number of hops, etc.). The bandwidth of the selected path should be the maximum one (among several paths having the same start and end nodes) but evaluated at the bottleneck link of that path. Additionally, the path might be constrained, e.g.: the number of hops (i.e., domains), should be lower than a maximum. For server status, one could consider the server proximity to the user, or server load. The MCDA algorithm has the quality that it can use several decision variables and make a global optimization.

For the path selection one may apply: a. *Single path between server and user* (usually provided by the current Internet routing based on minimum number of hops); b. *Random path selected among equal costs paths* between server and user, given that a multipath protocol is applied (e.g., modified Dijkstra algorithm); c. *best path among several paths* having similar costs in a defined range.

For server selection one may apply: 1. *Random selection*; 2. *Closest server* to the user (e.g., considering as metric the number of hops, i.e., domains - between server and user); 3. *Least loaded server* (the load can be evaluated as the current number of connections, or partially equivalent- as the total bandwidth consumed at the server output).

Considering combinations of the above factors, nine Use cases (and corresponding algorithms) can be defined: a.1, a.2, ...c.3, if independent decisions are taken for path, and respectively the server, with no MCDA algorithm. However, we will consider a global optimization MCDA algorithm with several decision variables taken from the above.

C. Two phases path-server selection procedure

The following simplifying assumptions are considered valid for this first version of the selection procedure:

- All servers are managed by the unique module called *Resource Allocator (RA)* belonging to SP Manager. The RA knows each server status, including its current load (number of active connections and bandwidth consumed at the server output). A degree of content-awareness exists in RA; it knows the inter-domain graph, and inter-domain link capacities.
- Each domain is considered as a node in the network graph, i.e., the intra-domain transport is not visible. This is a major realistic assumption in simplifying the amount of knowledge supposed to exist at SP level.
- All servers and users location are established offline, and are fixed. However, the system can accommodate the end user terminal mobility, given that in the content delivery phase a content server switching is possible.

- The total number of content objects (Max_no_CO) are distributed (offline mode, by an external caching process, out of scope of this algorithm) to server groups and between the servers of a given group, while the number of COs in a server should be $\leq Max_no_CO_per_Server$.
- The content object instances replicated in surrogate servers are known by the RA. A data structure CO_SRV_map contains the mapping of CO replica on servers. Each CO is stored in 1, 2, ..., K servers; K = maximum number of servers to replicate a content object.
- The time-life of a CO instance in a server is unlimited.
- All COs are delivered in unicast mode, so a "connection" is 1-to-1 mapped to a content consumption session. The COs have the same popularity.
- Each CO user request asks for a single CO; however, the same CO can be consumed simultaneously by several users, by using private connections.
- RA treats the User requests in FIFO (queue named $COreq_Q$) order.
- RA accepts or rejects user requests. Rejection happens if there are no servers, or no transport resources available. No further negotiation between the User and RA is assumed after a request transaction processing.
- The bandwidth occupied by a connection is equal to Bw_CO (in the first approach it can be considered constant). More generally this bandwidth is random, in a range $Bw_CO \pm \Delta Bw$.
- A connection load for the server and path will be Bw_CO , during Tcon interval measured from the connection request arrival instant (we neglect the processing time for content/connection requests).
- RA uses the most simple additive bandwidth management (no statistical multiplexing is assumed).
- The average duration of a connection (for content consuming) is Tcon. The real duration could be in a range TCon $\pm \Delta Tcon$.

Description of Phase 2 in pseudo-code

Request analysis and resource allocation (pseudocode)

// It is assumed a time process, which triggers activation of the main procedure, at each generic time tick instant Tk. This approach can serve also for managing the time lives of connections. The algorithm description is given below.

Each Tk

```
{ While COreq_Q non-empty
  {req = Extract_first_element_from COreq_Q( );
  Process_request (req); //processes the first request from the request queue COreq_Q
  Adjust_time_life_of_connections_in_servers; }
}
```

Process_request(req) // description of a user request processing

```
{Identify_Server_groups_and_individual_servers_able_to_provide_CO; // candidate servers for requested content
  //Search in the CO_SRV_map, by using the CO index in the request}
Create_candidate_servers_vector; // containing one entry for each such server
Collect_status_of_each_server; //from a data structure Server_status, the status of each sever is loaded in the
```

The Phase 1 (offline) general objective is to compute, on the inter-domain graph, (multiple) paths from server domains to user domains. No traffic load consideration is applied. The input data are: topology, inter-domain link capacities, location of servers, and users. Some constraints can be applied, e.g., bottleneck bandwidth (BB) on any path $\geq Bmin$; number of hops (domains) on any path $\leq NHmax$. The simplest metric is the classic one (number of hops).

More powerful approaches compute multiple paths: equal cost paths, or sets of paths having costs in a given range. Having more than one path would provide several MCDA choices opportunity. The multiple paths can be computed, by running a modified version of the classic Dijkstra algorithm [22]. A "better" (from QoS point of view) additive metric is: $link_cost = 1/B_{link}$, where B_{link} is the link bandwidth/capacity). Given that routing process is a classic one, it will be not detailed in this paper. The Phase 1 output is a set of sub-graphs, each one containing the multi-paths from a given group of servers to a given group of users. The Phase 1 algorithm is convergent. Its order of complexity is not higher than for different variants of Dijkstra based algorithms [22].

Phase 2

The Phase 2 of decision process jointly selects (for each user request arrived at RA), the best pair server-path (based on dynamic conditions) from the available candidates computed in the Phase 1. The signalling details user-RA are out of scope of this paper. The RA applies an admission control decision, followed/combined with an MCDA algorithm. The Phase 2 dynamicity means updating the paths and server loads according to the new requests arrived. Also considering the time-life of a connection, different server status items are updated when the connections are terminated. Note that there is no problem to downgrade the algorithm if complete path information is missing. More generally, the number of decision variables and the amount of information existent on them (static and/or dynamic) are flexible items in MCDA approach.

A description of Phase 2 is given below in a free-style simplified pseudo-code format.

```

// vector; in the most simple variant the status is: the current number of active
// connections for that server
Determines_sub-list_of_paths_for_each_candidate_server; // from the list of updated paths, by using information
// from the Phase 1
Create_candidate_list_of_path_server_solutions; //each solution is characterized by server load, bandwidth
// and number of hops
Delete_full_loaded_servers; //optional; it can be included in MCDA algorithm
Delete_elements_from_the_list_of_paths_associated_to_the_candidate_list:// optional; it can be done by MCDA
// those which have number of hops > NHop_max
// those which have Available Bandwidth < Bmin

Run_the_MCDA_reference_level_algorithm; // determine best path-server solution; policies can be included here

If_successful
then
  {Increase_success_list_statistics;
  Update_the_allocated_server_load;
  // Increase the number of active connections
  // Load & start timer associated to the time-life of this connection
  Add_additional_bandwidth_consumed_to_the_allocated_path_load_on_all_links;}
else increase_the_reject_list_statistics;
}
Adjust_time_life_of_connections // delete the terminated connections from the server status
For_each_server //Sv1, ...Svn
{ For_each_timer
  { If Active_flag=1 and Timer_value >0
    then {Timer_value --;
    If Timer_value = 0 then {Active_flag=0; NCO_srv --; }}}
Generation_Request_for_Content_object_
Initialization: TReq = random [1,...P*Tk];
Each Tk // equivalent with periodic interrupts at Tk seconds interval
{TReq = TReq - 1;
If TReq =0 then
  {k = random [1, .... Max_no_CO];
  Put_CO_req (User_id, Tcon, COk,)_in_COreq_Q;//place a new request in the queue
  TReq = random [1,...P*Tk]}; // restart timer and select a random interval until the
// next request generation}

```

V. SIMULATION SCENARIOS AND SAMPLES OF RESULTS

Considering the assumptions presented in the previous section and optimization algorithm, several simulation scenarios have been elaborated and experimented. This section presents a summary of results obtained. The objective is to determine the degree of effectiveness that an algorithm as MCDA can have versus other more simple selection decisions.

A. Simulation Framework

Figure 6 shows the multi-domain network model used in simulations. The network topology is organized on three tiers: Domains 5 and 6 belong to Tier1; Domains 3, 4, 8 and 9 belong to Tier2; Domains 1, 2, 7, and 10 belong to Tier3.

The general approach of the simulation framework is described below.

A *two phase selection* process is adopted. The prerequisites are: topology, EUs, servers are fixed; content object distribution is random but static;

Phase 1 executed offline, computes candidate paths from servers to users. *Phase 2* executed online, applies a selection algorithm (MCDA, etc.) and allocates resources; statistics are collected.

The EU requests are addressed to RA and are randomly distributed (uniform) in time. The RA searches a solution: runs a selection procedure (MCDA, etc.) to assign a (server, path) pair for this request; Updates the system status (servers, paths); counts the *success* and *reject* events in variable traffic load conditions

In order to compare MCDA results to other selection procedures the simulation model also supports:

Best server (BS) (least loaded)- single criterion :

1. get Request (*obj_id*) from an EU
2. select the set of servers having *obj_id*

3. select among them a BS (no matter the path)
4. Final decision: fail if *no_avail_Srv* or *no_avail_path*, *accept* otherwise
5. If, *accept*, then update the system status

Closest server (CS) single criterion :

- 1., 2., same as before
3. Select CS (no matter the path)
- 4., 5. Same as before

Scalar metric(COI) (Minkowski- order 1)

- 1., 2., same as before
3. Compute metric, select the solution having $M(1) = \min$
- 4., 5. Same as before

The Minkowski metric of order 1 [16] is computed as :

Cost = $NH/NH_{max} + SL/SL_{max} + PL/PL_{max}$, where NH = number of hops on a path, SL = server load, PL = path load.

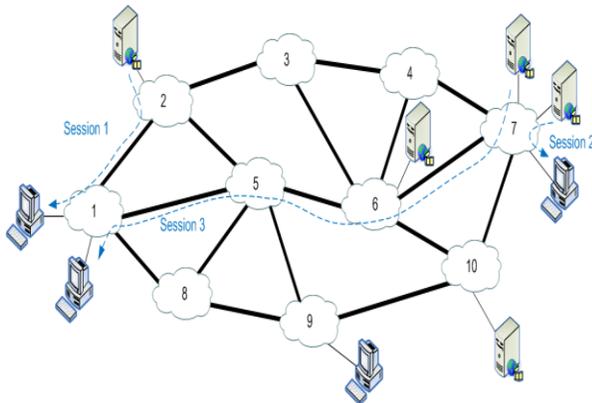


Figure 6. Network model used in simulations

B. Scenario 1 Results

Sample of the simulation results are presented below. The simulation model parameters are:

```

int Smax = 50; //Max no. of servers deployed over the
whole network
int Cmax = 100; //Max no. of original content objects
int D = 10; //The number of network domains within the
whole network
int Nmax = 300; //Max no. of user requests within the
simulation time interval
int T = 1000; //The simulation time interval
int M = 300; //The lifetime for a server-user multimedia
streaming session
int CD = 15; //Max number of content object replica
within whole network
    
```

There are several streaming servers on each domain. The users are requesting for different multimedia contents. For any accepted request, a new server-user session is opened.

Figures (7-10) show comparative results for different algorithms: the success ratio versus request rate. One has considered different replication factors, here samples are shown for versus $rf= 3, 8$ replicas. The following comments on results can be stated:

- When the system is significantly non-saturated (low request rate) different algorithms produce similar results. No significant gain is observed for MCDA.
- When the system saturation is very high, again the algorithms results start to be similar. Therefore, for a given network dimensioning one should evaluate the region (versus traffic load), in which more complex algorithms such as MCDA may produce significantly better results, versus trivial ones.
- When replication factor rf increases the MCDA results are clearly better. One can observe an improvement ratio of 40% in case of $rf= 8$.
- When replication factor rf increases the Minkowski-1 metric-based algorithm produces similar results as MCDA.

Note: The following diagrams (in Scenarios 1 and 2 of results) have the “request rate” label on oX axis. Actually, the values are the total request number per simulation time. However, the label was set with attribute “rate” to emphasize the fact that the request rate is increasing proportionally as the total number of request is higher.

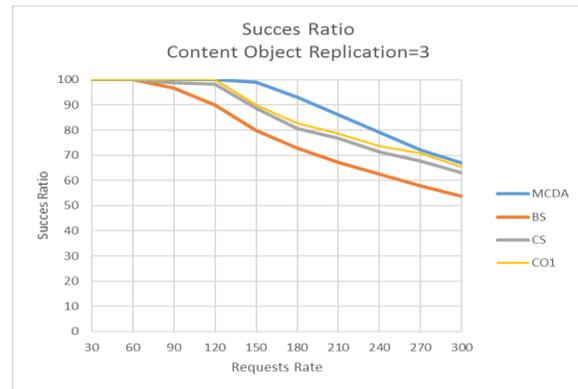


Figure 7. Success ratio for CO replication = 3

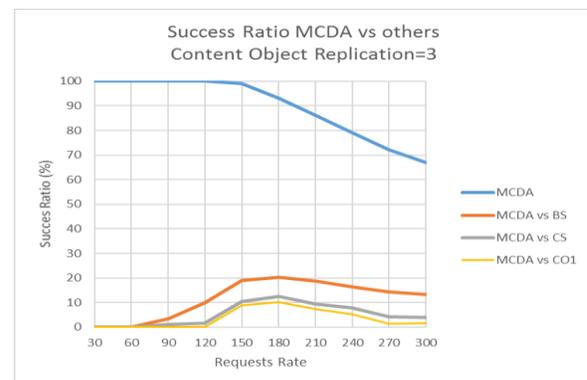


Figure 8. MCDA gain for CO replication = 3

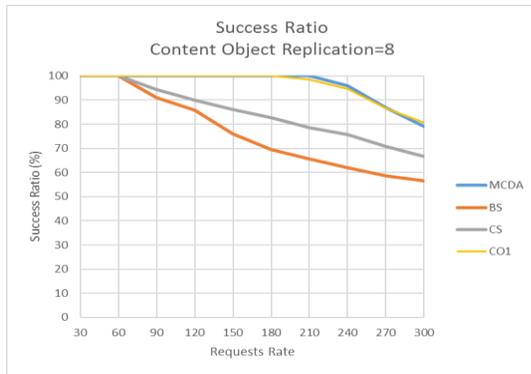


Figure 9. Success ratio for CO replication = 8

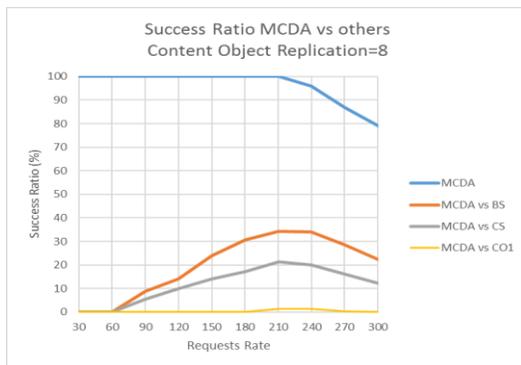


Figure 10. MCDA gain for CO replication = 8

Figure 11 presents a summary of experiments showing how MCDA success ratio varies when different replication factor is applied and different request rates exist. The surface is called “decision space”, because one can decide based on such surfaces when it is worth to apply a MCDA algorithm, and when other more simple approach could be used.

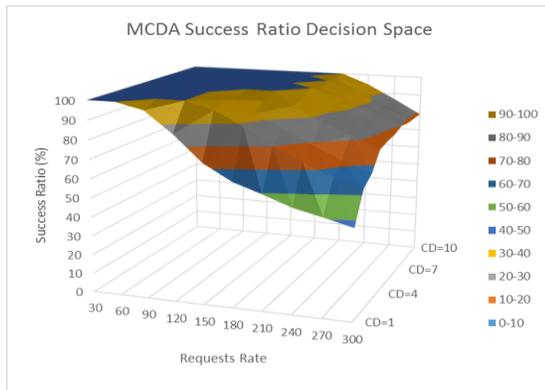


Figure 11. Summary of MCDA success ratio versus request rate and replication factor

Figure 12 serves to illustrate more the previous statement associated to Figure 11; it shows the MCDA improvement versus Best Server selection. It is again seen that MCDA can offer significant gain only if the replication factor is higher than 5 and for higher request rates.

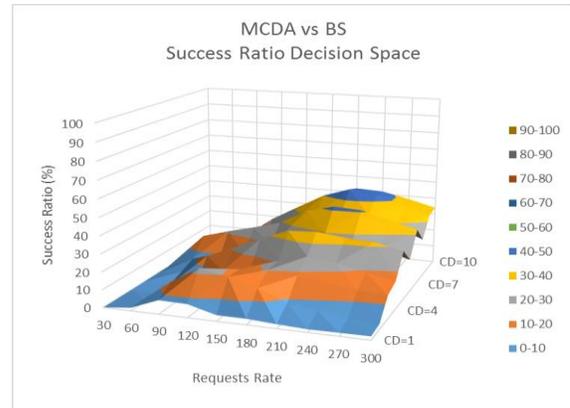


Figure 12. Summary of MCDA success ratio versus request rate and replication factor

C. Scenario 2 Results

The same network has been used. Some specific conditions of these experiments have been:

Simulation Interval $T=100$

Max number of requests $N_{max} = \text{variable} = [100 \dots 5000]$;

Rate = N_{max}/T

Multimedia session duration $M=30$

Link capacities: tier1=100, tier2=30, tier3=10

Number of replicas for the content objects:4.

Note: The diagrams below have been extracted from experiments where it is wanted to check the correctness of rejection reasons due to different non-available resources (either server saturation or link/network paths saturation). In order to emphasize such effects, an increasing additional load in the network has been generated. Instead of adding background traffic the method to increase the network load has been by not releasing in the network the paths occupied by a session after its termination (however, the server is freed). In this way, the network traffic is constantly increasing during the simulation time. Consequently, it is expected that a correct behavior of the algorithm will determine more rejections due to network links saturation.

Figure 13 shows in the above condition the performance of the Best Server method selection, given that the total traffic in the network is increasing (two reasons: request rate increasing and, the percentage of rejections due to path load is increasing). This shows that in real cases the network connectivity capacities should be enough high, otherwise a high number of servers does not help so much.

A similar behavior is exposed when the selection is performed based on Minkowski - order 1 metric- see Figure 14. Figure 15 shows that still MCDA has ~20% gain over Best-server method even at high request rate. However, the

gain is less between MCDA and CO-1 (Figure 16). The explanation of this is that CO-1 includes several parameters in its scalar formula.

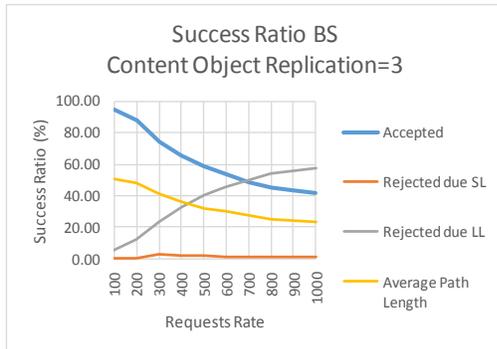


Figure 13. Best Server (BS)- performance

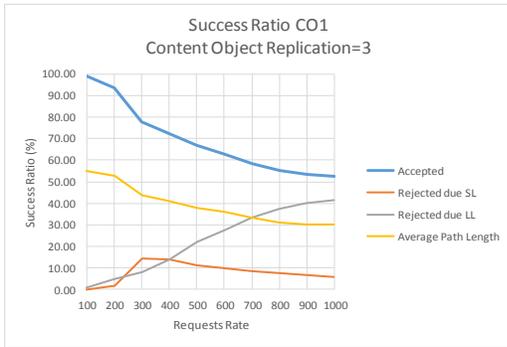


Figure 14. CO-1 - performance

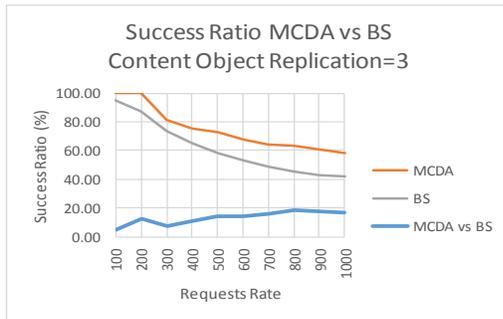


Figure 15. MCDA gain versus BS

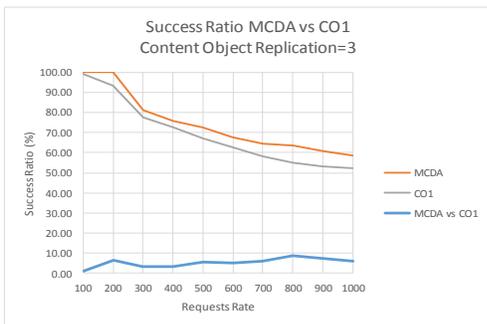


Figure 16. MCDA gain versus CO-1

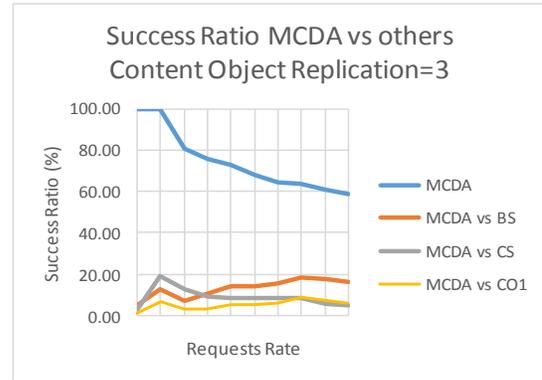
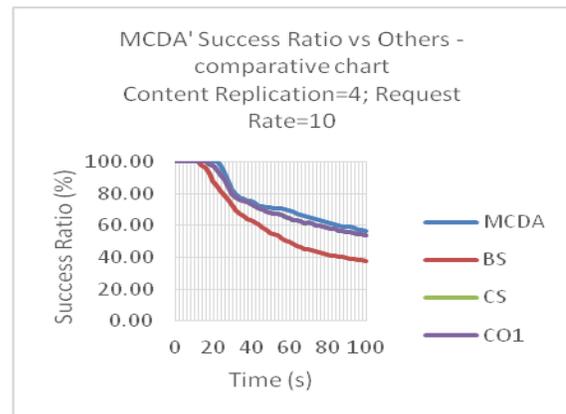
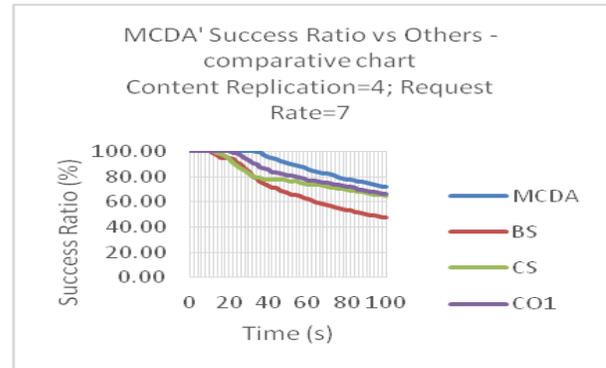


Figure 17. MCDA gain versus others

Figure 17 shows an aggregated diagram: MCDA gain versus other methods. It is seen the closest to MCDA is CO-1.

Figure 18 contains four diagrams representing the behavior of the system versus time. One can see that for a high request rate the system becomes saturated sooner and the MCDA does no more offer gain versus other selection methods (e.g. rate = 7 versus rate = 20, or 50)



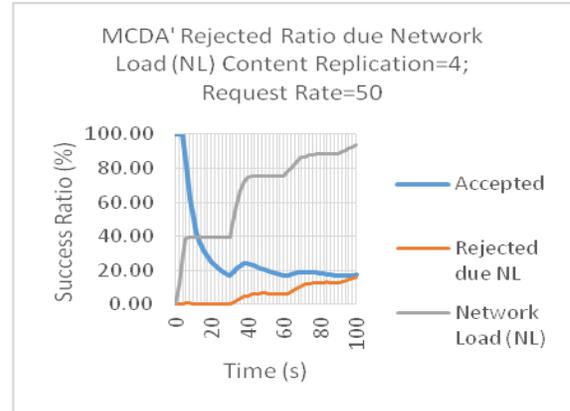
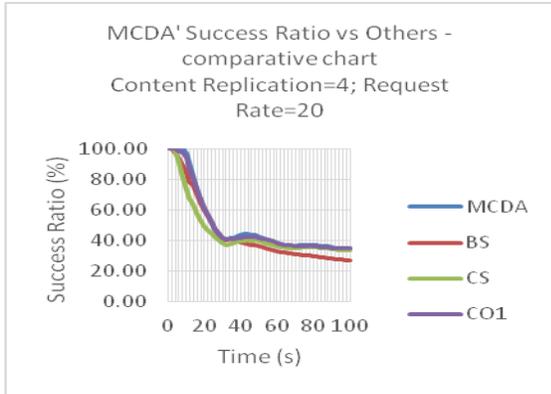


Figure 19. MCDA behaviour at high request rate

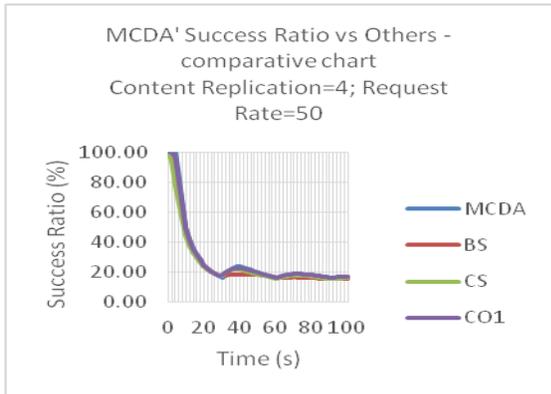


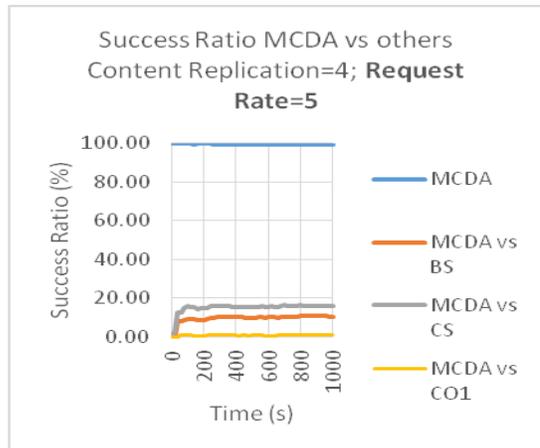
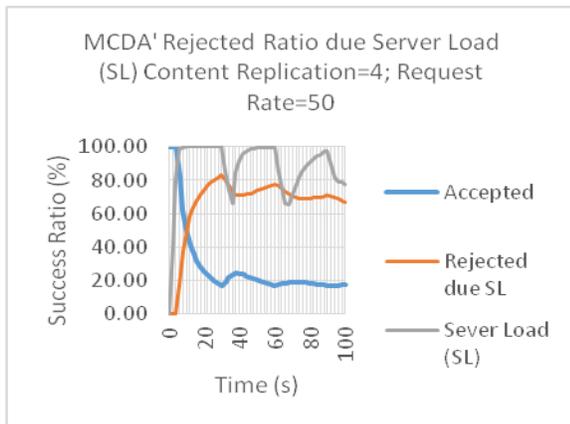
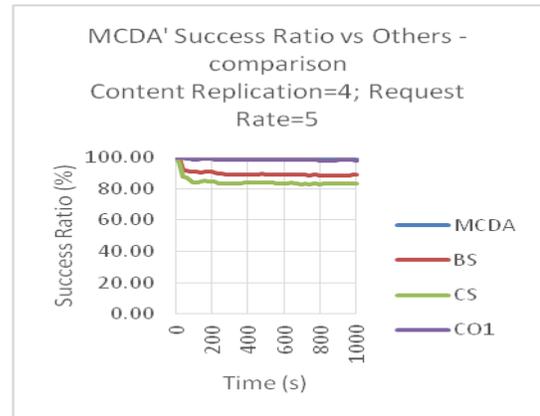
Figure 18. System behavior versus time for different request rates, going towards system saturation

D. Scenario 3 Results

The conditions of simulations are the same as in Scenario 2 except the fact that no other additional traffic is present in the network but only the media session traffic. Each session when terminated will determine release of the resources on all network links previously used.

The simulation time has been increased to allow the time diagrams to show the region of some stable conditions of load for servers and network paths.

At high request rate and rather equal session duration ($T \approx 30$ sec.) a cyclic behavior is observed (Figure 19). When the system is unloaded, a lot of requests are accepted in short time. Then the servers become saturated (100% load). The sessions terminate at $t \approx 30$ sec., all rather in the same short interval (compared to session duration) and Server load decreases sharply. Meantime new requests arrive (however, the network is partially loaded now) and the Server Load increases again until $t \approx 60$ sec, when a new decrease is observed and so on.



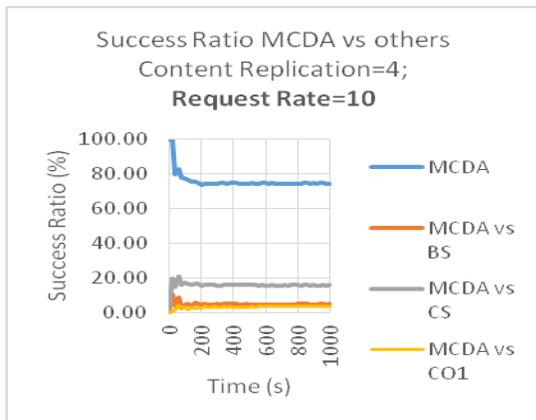
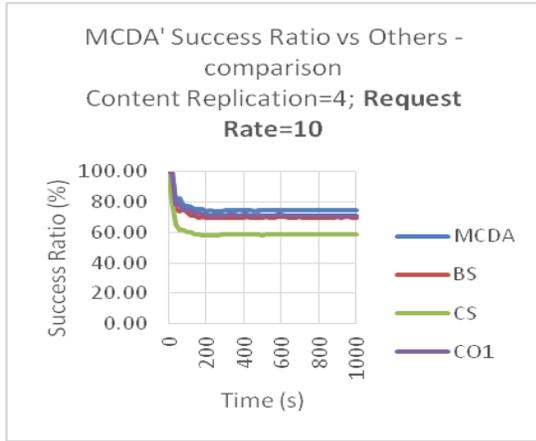


Figure 20. Comparison of MCDA behavior versus other algorithms versus time

The results presented in Figure 20 show a better performance of MCDA on long term, versus other methods. The MCDA algorithm is compared to Best Server, Closest Server and Minkowski 1- metric algorithm. The main conclusion is that MCDA and CO1 expose very close performances while BS and CS are weaker- due to either server or path non availability.

VI. POLICY GUIDING THE MCDA

Several remarks can be done related to the basic reference level algorithm:

- The formula $min_s = min\{v_{si}'\}; i = 1..m$ (3), selects as representative of each candidate solution, the “worst case” value, i.e., for all other variables/parameters, this solution has “better” normalized values then this representative. This is arithmetically correct, however, in practice, this “worst” case parameter might be actually less important than others, either from technical or business (i.e., policies) point of view.
- In some particular cases with dependent variables (e.g., delay/jitter) the solution selected could be not the most appropriate, from actual implementation point of view.

- The step 2 compares values coming from different types of parameters (e.g., 1/Bwdth, delay, jitter, server load, etc.) - independent or dependent on each other. The normalization allows them to be compared in the $max\{ \}$ formula. However, the numbers compared are from items having different nature. This is an inherent weak property of the basic algorithm.
- More important is that the SP might want to apply some policies when selecting the path-server pair for a given user. Some decision variables could be more important than others. For instance, the number of crossed domains (no_of_hops in MCDA) can be the most important parameter – given the transit cost. In other cases, the server load could be more important, etc.

A simple modification of the algorithm can support a variety of SP policies. We propose here a modified formula :

$$v_{si}' = w_i(r_i - v_{si}) / (r_i - a_i) \quad (3')$$

where the factor $w_i \in (0,1]$ represents a weight (priority) that can be established from SP policy considerations, and can significantly influence the final path-server selection. This will solve the above mentioned issues.

A sample example below shows the optimization obtained. Let us consider a selection scenario, in which the decision variables are given in Table I, and six candidates in Table II (entries are native not-yet normalized values)

Priorities are introduced in Table I, derived from SP policy. Here, the server load and numbers of hops are considered the most important.

One can define: $a_1 = 0, r_1 = 100; a_2 = 0, r_2 = 10; a_3 = 110, r_3 = 10; a_4 = 0, r_4 = 50; a_5 = 0, r_5 = 100.$

TABLE I. DECISION VARIABLES EXAMPLE

Decision variables	Semantics	Units	Priority
v_1	server load	(%)	1- max
v_2	number of hops	Integer	1
v_3	available bwth on the path	Mbps	2
v_4	jitter	ms	3
v_5	E2E delay	ms	4- min

TABLE II. CANDIDATE SOLUTIONS EXAMPLE

	S1	S2	S3	S4	S5	S6
v_{s1}	0	20	40	70	80	100
v_{s2}	5	7	6	3	4	5
v_{s3}	40	20	50	80	50	60
v_{s4}	0	10	30	20	10	30
v_{s5}	30	80	70	40	30	50

Applying the basic algorithm (i.e., with no priorities) simple computation will show that formula (4) is $max\{0.3, 0.1, 0.3, 0.3, 0.2, 0\}$, showing that solutions s1, s3, s4 are equivalent. However, examining the initial input candidate values, it is clear that s1 is the best (server load=0, and sufficient available bandwidth- compared to others).

Now, we introduce policies, assuming the priorities assigned in Table I. Some weights (acting as compression factors) can be defined, e.g., $w_1 = 0.5$, $w_2 = 0.5$, $w_3 = 0.7$, $w_4 = 0.8$, $w_5 = 1.0$. Then applying the formula (3'), one gets a new set of values for the formula in (4), i.e., $\max \{0.21, 0.07, 0.2, 0.15, 0.1, 0\}$. It is seen that s1 solution is now selected as the best, which corresponds to the intuitive selection of it.

Some other examples have been checked to verify the prioritized selection capability of the modified MCDA. Note that despite its simplicity the modification proposed can have major impact on algorithm results, given that different SP policies can be defined, depending on user categories, content server exploitation needs, networking environment, etc. Therefore, the weighting factors in practice do not come from some formulas, but should be chosen, based on the defined priorities of the SP. A natural usage of the modified algorithm proposed here could be to select several sets of best solutions, fit to the different policies of the Service Provider.

VII. CONCLUSIONS AND FUTURE WORK

This paper presented a study on multi-criteria decision algorithms and procedures for best path-server selection in a content delivery system.

While applying some previous ideas of two phases procedure (offline and online) the solution adopted here is a flexible (supporting many use cases) modified decision procedure, which additionally can capture some policy related priorities for decision variables. It was shown that such modifications can enhance the added value of the decision taken by the algorithm.

Future work will be done (in the DISEDAN project effort) to simulate the system in a larger network environment, and finally, to implement the described procedures in the framework of a system dedicated to content delivery based on a light architecture.

ACKNOWLEDGMENTS

This work has been partially supported by the Research Project DISEDAN, No.3-CHIST-ERA C3N, 2014- 2015 and project POSDRU/107/1.5/S/76903.

REFERENCES

- [1] E. Borcoci, M. Vochin, M. Constantinescu, J. M. Batalla, and D. Negru "On Server and Path Selection Algorithms and Policies in a light Content-Aware Networking Architecture," ICSNC 2014: The Ninth International Conference on Systems and Networks Communications http://www.thinkmind.org/index.php?view=article&articleid=icsnc_2014_1_40_20077
- [2] <http://www.chistera.eu/sites/chistera.eu/files/DISEDAN%20-%202014.pdf>, retrieved: 07, 2014
- [3] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A Survey on Content-Oriented Networking for Efficient Content Delivery," IEEE Communications Magazine, March 2011, pp. 121-127.
- [4] V. Jacobson et al., "Networking Named Content," CoNEXT '09, New York, NY, 2009, pp. 1-12.
- [5] J. Pan, S. Paul, and R. Jain, "A survey of the research on future internet architectures," IEEE Communications Magazine, vol. 49, no. 7, July 2011, pp. 26-36.
- [6] P. A. Khan and B. Rajkumar, "A Taxonomy and Survey of Content Delivery Networks". Department of Computer Science and Software Engineering, University of Melbourne, Australia: s.n., 2008. www.cloudbus.org/reports/CDN-Taxonomy.pdf.
- [7] FP7 ICT project, "Media Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments," ALICANTE, No. 248652, <http://www.ict-alicante.eu>, Sept. 2013.
- [8] <http://wp2.tele.pw.edu.pl/disedan/>, retrieved: 07, 2014.
- [9] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering". Struct Multidisc Optim Eds., No. 26, 2004, pp. 369-395.
- [10] J. Figueira, S. Greco, and M. Ehrgott, "Multiple Criteria Decision Analysis: state of the art surveys," Kluwer Academic Publishers, 2005
- [11] J. R. Figueira, A. Liefoghe, E-G. Talbi, and A. P. Wierzbicki "A Parallel Multiple Reference Point Approach for Multi-objective Optimization," "European Journal of Operational Research" 205, 2 (2010), pp. 390-400, DOI: 10.1016/j.ejor.2009.12.027.
- [12] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization". Lecture Notes in Economics and Mathematical Systems, vol. 177. Springer-Verlag, pp. 468-486.
- [13] T. Kreglewski, J. Granat, and A. Wierzbicki, "A Dynamic Interactive Decision Analysis and Support System for Multicriteria Analysis of Nonlinear Models," CP-91-010, IIASA, Laxenburg, Austria, 1991, pp 378-381.
- [14] J. M. Batalla, A. Beben, and Y. Chen, "Optimization of the decision process in Network and Server-aware algorithms", NETWORKS 2012, October 15-18 2012, Rome.
- [15] A. Beben, J. M. Batalla, W. Chai, and J. Sliwinski, "Multi-criteria decision algorithms for efficient content delivery in content networks," Annals of Telecommunications - annales des telecommunications, vol. 68, Issue 3, 2013, pp. 153-165, Springer.
- [16] E. Borcoci, ed., et al., D2.1 "System requirements and comparative analysis of existing solutions for media content server selection and media adaptation," July 2014, <http://wp2.tele.pw.edu.pl/disedan/>
- [17] C. Liuy, R. K. Sitaramanyz, and D. Towsley, "Go-With-The-Winner: Client-Side Server Selection for Content Delivery," <http://arxiv.org/abs/1401.0209>, retrieved: 07, 2014.
- [18] J. Mongay Batalla, C. X. Mavromoustakis, G. Mastorakis, D. Négru, and E. Borcoci, "Evolutionary Multiobjective Optimization algorithm for two-phase content source selection process in Content Aware Networks," submitted to Springer 4OR - A Quarterly Journal of Operations Research.
- [19] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," MultiMedia, IEEE, vol. 18, no. 4, pp. 62 - 67, 2011.
- [20] ETSI TS 126 247 V11.7.0 (2014-07) (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH), (3GPP TS 26.247 version 11.7.0 Release 11, 2014).
- [21] O. Oyman and S. Singh, "Quality of Experience for HTTP Adaptive Streaming Services," IEEE Communications Magazine, April 2012, pp.20-27.
- [22] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to Algorithms," The MIT Press, Cambridge, Massachusetts, 2000, ISBN: 0-262-53091-0.

User Preferences and Segments in App Store Marketing: A Conjoint-based Approach

Stephan Böhm, Stefan Schreiber, and Wendy Colleen Farrell

RheinMain University of Applied Sciences

Department of Media Management,

Center of Advanced E-Business Studies (CAEBUS),

Wiesbaden, Germany

e-mail: {stephan.boehm, wendycolleen.farrell}@hs-rm.de, stefan.schreiber87@web.de

Abstract—Today, little is known about how the various elements involved in the presentation of mobile applications (apps) in app stores influence the download or purchase decision of potential users. Current publications primarily focus on the possibilities and technical tools of app store marketing based on best practices or experience. However, research on customer preferences with regards to the presentation of apps in app stores as well as the impact of single app store elements on purchase or usage decisions has yet to be addressed. In this context, the key research objectives of this paper are to not only analyze the impact of individual app store elements on customer choice but also see if the customers can be segmented into homogenous groups according to their preferences. Accordingly, this study will identify the relative importance of individual app store elements, from both a general or mass market perspective as well as a user segmentation perspective, and derive recommendations on how to successfully present mobile applications in app stores. With this objective in mind, a conjoint analysis of a fictitious mobile messaging app in the Apple App Store was carried out for the purpose of identifying the relative importance of app store elements in the mass market. It was followed by a Latent Class Analysis, which looked at how those preferences differed among different market segments.

Keywords—Mobile App Marketing; App Store Elements; App Marketing; Consumer Preference; Conjoint Analysis; Market Segmentation; Latent Class Analysis.

I. INTRODUCTION

As discussed in [1], the number of mobile applications is steadily growing. More than a million applications are now available for Android and iOS in the respective app stores (i.e., GooglePlay and Apple App Store). Accordingly, the competition among individual app providers is constantly rising [2]. It has long since ceased to be enough to simply turn a good idea into an app. More and more, the question has become, which factors trigger the user's purchase decision. Numerous managers in the mobile phone business are now forced to deal with this situation and to define mobile app marketing strategies on how to achieve and defend a competitive position for their apps in the market.

Marketing plans and strategies are usually created according to the concept of the marketing mix, which also plays a key role in mobile app marketing [3]. The marketing mix should be an optimal combination of marketing tools from the areas of Product (product policy), Price (pricing policy), Pro-

motion (communication policy) and Place (distribution policy) [4]. These "4Ps" are also the components of the app store marketing toolkit. Product policy starts at a very early stage and deals with the app idea and with the subsequent design of the application [3].

With regards to pricing policy both before and after the launch of the mobile app, a wide range of decisions have to be made. These decisions range from adequate price level to dynamic pricing strategies designed to systematically alter prices over time in order to react to changes in actual demand and current market conditions. However, pricing policy is limited by the possibilities and restrictions of the app stores. For example, the app stores may specify certain price points to be selected or not permit providers to offer trial versions for a limited period of time [5].

Distribution policy generally deals with all the marketing decisions and activities concerned with the delivery channel from the producer to the customer and therefore from production to consumption [4]. As early on as the development stage of an app, the distribution channel is determined, or at least influenced, by the technical implementation. So-called web applications, for example, can simply be made available for download per link or published via any webserver. The distribution channel for so-called hybrid and native applications, on the other hand, is the app store. Before use, they must be completely downloaded and installed on the mobile device. While native applications are created using the platform-specific development environment and programming language, web technology is usually used with hybrid applications. Additional development frameworks and tools, however, allow for further processing and compilation of this source code in a way that enables its distribution via an app store in a similar way to a native application.

Within the communication policy, we have to differentiate between activities inside and outside the app store. This includes advertising and other activities, which provide and disseminate information aimed at familiarizing the potential customer with the app and its features. App stores are usually the only official channel for users to buy and install new apps on their mobile devices. Thus, the communication policy within the app stores and the corresponding design of the various app store elements are of particular importance [6]. Here, it must be noted that each store has its own specific regulations and guidelines on how to publish an app for distribution as well as which elements can be used to present the app in the store.

However, although the regulations vary in detail, the core concepts and the core elements for the app presentation are quite similar.

All the aforementioned app marketing activities need to be aligned with the intended target group of the app. Hamka et al. argue that the mobile market is evolving at an ever increasing rate as is the behavior of mobile users. Accordingly, they suggest that it is an imperative to segment the market “i.e. divide the addressable market into segments that have a consistent demographic, psychographic or usage pattern” [7]. However, the options for segment specific app marketing are limited. Currently, when a mobile app is launched the app elements can be adapted to local markets by selecting national stores and providing market specific app elements (e.g., app descriptions in different languages). In addition, the app can be placed in categories that represent the general usage concept of the app. These categories range from educational to games to tools. That being said, while these groupings help the potential user search of a specific type of application, they do little to actively market to specific groups of potential users. This might be sufficient for standard applications targeting relatively homogenous user preferences found in mass markets. In contrast, user groups with a differing preference structure looking for non-standard apps would argue for a more segment specific approach in app store marketing.

In this context, the objective of this study is to first develop appropriate recommendations for the setup and design of important app store elements, to empirically validate common app store marketing best practices and to determine potential user groupings based upon preferences. For this reason, a conjoint approach was chosen to analyze user preferences and characterize the relative importance of different app store elements. Then, using the data collected as part of the choice based conjoint analysis, a latent class analysis was conducted. This analysis was used to discover user groups with similar preference structures according to the presentation of app store elements.

With this in mind, Section II presents a short discussion on related work and current best practices in app store marketing. Section III describes important elements of the presentation for mobile applications in app stores. The explanations refer to the example of the Apple App Store; can, however, to a great extent be generalized to include other app stores. In Section IV, the methodological approach of this study is then described. Significant results of the conjoint analysis and the subsequent latent class analysis are presented in Section V, before we finally discuss the central findings and recommendations for practical implementation in the concluding section.

II. RELATED WORK

Mobile app marketing is still a relatively new marketing topic. It was not until the first app stores emerged that the necessity for a market-oriented way of thinking when developing and marketing mobile apps started to become apparent [5]. In principle, we can say that many well established concepts from general marketing practices are transferable to mobile app marketing. Consequently mobile app store marketing

adopts standard marketing principles and tools and adapts them to the needs of the app specific market.

Current literature on mobile app marketing predominantly focuses on guidelines and recommendations for the successful monetization of app concepts. For example, the topic of app marketing can be found as part of the technical literature on app development in which the monetization of the app in the app store is seen as being the final step in the app development process [5][8][9]. Additionally, more specialized publications focusing on mobile app marketing are available as well [2][6][10][11]. However, most of these publications comprise structured guidelines and extended checklists on how to successfully monetize mobile applications based on the authors’ experience or the discussion of successful case studies. In contrast, scientific research on app stores and app (store) marketing is rather rare today. Only few publications have so far dealt with individual aspects of app stores, mainly focusing on app ranking mechanisms and fraud [12][13][14], pricing strategies [15] or recommendations and user reviews [16][17].

Against this background, a significant research gap can be observed with regard to the availability of empirically based recommendations on the market-oriented configuration of app store elements. The suggested research approach, a study measuring customer preferences and segments based on a conjoint analysis, has been applied to software selection processes and even to mobile application development [18][19], but is rather new to the specific area of app store marketing. Accordingly, this study will attempt to answer the following two research questions:

- What are the most important app store elements from a user perspective and how should those elements be presented (based on the example of a messenger app)?
- Do users of mobile messenger apps (in Germany) fall into specific segments based on their preference structures for the presentation of app store elements?

Understanding the answer to these questions can help developers and marketers to more effectively reach the intended end user and communicate the product benefits.

III. APP STORE ELEMENTS

As stated above, the design of the various app store elements is one of the key instruments of mobile app marketing. Potential users search for suitable mobile applications in the app store and obtain information about their features and properties [3]. In order to acquire a common frame of reference for this study, we focused solely on the Apple App Store. There are various app stores for different mobile operating systems, which are characterized by different appearances, but which are fundamentally similar in terms of the possibilities to present mobile applications.

A fictitious messenger app was chosen to concentrate on the importance of the app store elements and prevent participants from being biased by earlier purchase decisions, knowledge of real-world app presentations or brand preferences. The Apple App Store can be accessed via several mobile devices. It is possible, for example, to open the app store

via smartphones (iPhone) and tablets (iPad) to download applications. However, the number of elements is the same for all devices and always identical in each case.

In total, based on an analysis of the Apple App Store and best practices derived from the mobile app marketing literature in Section II, eight key app store elements were examined for this study, which will be described in more detail below. Moreover, the study also deals with variations of each of the attributes, which were compared and examined with regard to their influence on customer preference in terms of a purchase or usage decision. The fictitious messenger app was presented to the participants of the study based on the attributes and its selected attribute levels only. There was no prototype or trial-version in an app store available in this study.

A. App Icon

The app icon is seen as being one of the most crucial elements, as it is generally the first visual element that a potential user sees. The purely aesthetic design of the app icon can already have an effect on the development of user preference, for example in the way that the icon makes an impression and is taken as an indication of the quality of the app. The app icon and the app name are central design elements in many app stores, not least because they would be the first items that appear on the search results page [5][20]. In Figure 1, three icon versions are shown that were developed for a fictitious messenger app in the study.

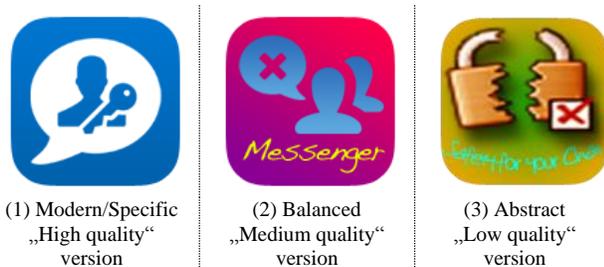


Figure 1. App Icons Variations

In the form of these icons, the intention is to refer to a particular messaging app, which is characterized by an especially high level of security. Best practice guidelines have been used to develop the design variations [21]. For example, the coloring and the legibility were varied in order to portray the spectrum from a representative “high” to a “low” quality design. The same is also true for the clarity of the graphic elements to visualize the messaging and security features of the app. While icon (1) has easy to understand graphical representations of messaging, icon (3) uses a vague illustration and faint writing. The consideration of the icon design as an attribute will allow an empirical verification of the aforementioned existing best practices in the study.

B. App Name

As mentioned above, the name of the app is also a central element with respect to the presentation of mobile applications in app stores, as it is shown in the app store’s search and rank-

ing lists and may therefore influence the user’s purchase decision [5]. The app name should fulfil certain criteria in order to be easy to remember on the one hand, and easy to find via the app store’s search algorithms on the other. Ideally, solutions to internationalize the name should also be available [3]. For the test app in the conjoint analysis, the same name was used for all three, but a claim was added for extra clarification. The claim varied from a simple allusion to security to a technical description, which is difficult for the average user to understand (high to low comprehensibility):

- „high“: SafeTalk – Your Safe Messenger
- „medium“: SafeTalk Secure Messenger
- „low“: Safetalk with AES-256 Encryption

C. Reviews („stars“) and the number of reviews

The reviews in the app store are assigned according to the star principle (1–5 stars) and are – together with the number of total reviews – an initial indicator for the user of how satisfied other users were with the app after downloading. A high number of stars is perceived as being a positive purchase recommendation [6]. App providers should note that star reviews are not immediately displayed for new apps but are only published once a meaningful average value can be calculated. In the Apple App Store, this means a minimum of 5 reviews. Apple also differentiates according to countries. At present, it is not possible for the user who is giving the review to interact directly with the app provider [3]. The following analysis includes the review alternatives none, three and five stars.

D. Price

Pricing is another element that is immediately displayed on the search result page and in all the app store’s lists (for example in the „top charts“) and can therefore influence the user’s purchase decision during the app selection process. For the analysis in this study, a cost-free version and three price points were chosen, which represented a low, a medium and a high price segment, respectively, in comparison to actual mobile messaging applications (0.89 EUR, 1.79 EUR, 2.69 EUR).

E. Screenshots

Screenshots are usually only visible in the detail view of an app, with the exception of the result page of the search feature. Here, the first of a total of five possible screenshots is already shown in the preview. Screenshots have several tasks: On the one hand, they should display the features of the mobile application and, on the other, communicate the app’s design [3]. Screenshots offer crucial support to the descriptive text as many users do not read this or only read it in part and therefore rely heavily on the screenshots for their purchase decision [20]. App store users draw conclusions from the screenshots as to the aesthetics and user friendliness of the mobile application as a whole [6]. In this study, three different qualities of screenshots were created (high, medium, low), which vary with regard to recognisability and clarity of the functional elements of the mobile messaging app. The functional “low quality” screenshot, for example, displays purely func-

tional content, whereas the notated “high quality” one highlights important core functions with accompanying explanations.

F. App Description

The descriptive text is the only element presented here, which appears solely in the detail view of an app once it is opened. The Apple App Store allows a descriptive text with a maximal number of 4000 characters [6]. The descriptive text is important for two reasons: Firstly, potential customers are presented with a list of sales arguments and secondly, the search algorithms of most app stores use the text to carry out corresponding search requests. As the optimization for search purposes was not the main focus here, the quality of the descriptive text was varied mostly in terms of comprehensibility. Here again, three levels of quality were created (high, medium, low). Whereas the user oriented “high quality” description used simple language and comprehensible wording, the complex “low quality” descriptive text was characterized by technical terms, which the average user would find difficult to understand. In addition, the text was automatically translated as is often the case in app stores, which reduced the comprehensibility yet further.

G. Server Location (as an additional attribute)

As a messenger with special focus on secure communication had been chosen as a fictional product for analysis, an additional attribute entitled “server location” was included in the study for evaluation. This is not an element of an app store in a narrow sense, but an important company related attribute of the app provider that can be emphasized within the app description. While the aforementioned attribute is used to measure how the quality of language influences user preferences, the server location is an example of how various app characteristics, even if just mentioned in the description, could have an impact on customer choice. Due to current discussions about data security in Germany [22], heightened customer awareness was assumed to be a significant influencer on customer preference. The goal of including this attribute was to test whether and to what extent such attributes contribute to the user’s purchase decision in comparison to the other marketing-related app store elements. Server locations in the US, in Germany and an unknown server location were included in the study.

IV. METHODOLOGY

In identifying the most appropriate methodology to analyze the app store elements and provide potential user segmentation, multiple methods were considered. Conjoint analysis was identified as the most appropriate method to analyze the user preferences for the various app store elements. Based upon that decision, Latent Class analysis was selected to analyze the potential segmentation of users based on their preference structures. In the following, these methods and the reasoning for applying them in this study are discussed in detail.

A. Conjoint Analysis

The conjoint analysis is considered to be the standard method when investigating customer preferences and buying

decisions. Traditional Conjoint Analysis (TCA) goes back to the year 1964 and was developed by the psychologist Luce and the statistician Tukey [23]. TCA, as well as all the subsequent versions of conjoint analysis, basically deals with the measurement of preferences for product attributes. Instead of asking the participants directly about the importance of attributes, conjoint analysis is based on the evaluation of product profiles. Each product profile consists of several attributes describing the product characteristics (e.g., brand, price, design, etc.). Different product profiles are derived by variation of attribute levels (e.g., high, medium, and low price). An analysis is always carried out in such a way that each product profile or “stimulus” has to be examined and assessed from a holistic perspective or *considered jointly*) [24][25]. Instead of asking directly about the importance of a product attribute, conjoint analysis considers products as bundles of attributes, on which the customer decides and makes trade-off decisions. The approach is better aligned to real-world purchasing decisions and the part-worth utilities of the attributes can be decomposed by using statistical methods like regression analysis.

For this reason, the conjoint method is well suited to analyze the impact of different app store elements on the customer choice decision. As a result, the relevance of the key app store elements, derived from the practical literature, can be empirically validated based on the example of fictitious messenger app. The analysis also provides the relative importance of the different app store elements for market success. From a more practical perspective the results could be used by an app provider to determine the optimal app store configuration for the analyzed secure messenger app or to conduct market simulations based on different configurations. However, the study at hand focusses on the relative importance of the app store elements. The reference to a fictitious messenger app was required only because the conjoint analysis cannot be conducted based on a non-specific and generic “mobile app”.

Since the mid-sixties, conjoint analysis research has evolved and produced several variants that can be divided into traditional and more recent approaches. Traditional Conjoint Analysis (TCA) can be applied by using trade-off or full-profile approaches but its significance in research has been declining since its first appearance due to limitations on the number of attributes as well as other methodological and statistical problems [26]. Of the more recent approaches, Choice Based Conjoint Analysis (CBC) and its variant, the computer-aided Adaptive Choice-Based Conjoint Analysis (ACBC) are taken into consideration for this study.

CBC is the most popular conjoint analysis today. In CBC, unlike TCA, discrete selection decisions are analyzed instead of preference decisions [27]. During CBC, the subject is therefore not asked to make an order of precedence of all the product profiles, but must select the preferred product profile within a set of alternatives or, if such an option is included, reject the choice by deciding on a “none option” [24][25] as shown in Figure 2. The ACBC is a computer-aided enhancement of classic CBC and includes an adaptive approach. This means that every piece of information supplied by the test subject during the course of the interview gradually reveals the formation of his/her preference structure so that the questions posed to him/her can be successively adapted to the answers

[28]. In this context, the first consideration for the study was to determine, which kind of conjoint analysis should be applied. For best results, CBC is recommended if the product bundle in question has around six attributes or less, however, the method can be carried out with up to ten attributes. ACBC has proved to be especially suitable if 5 to 15 attributes are to be examined. However, it is characterized by a more complex and time-consuming questioning process [29].

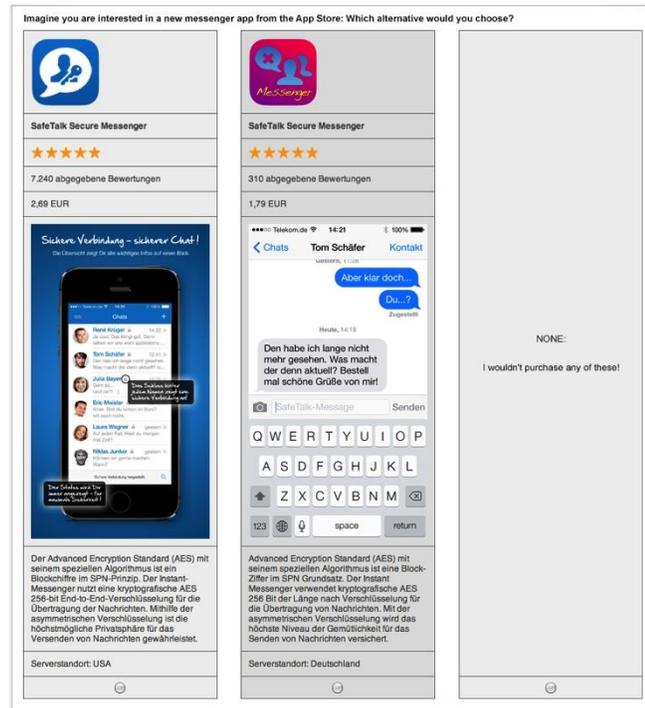


Figure 2. Example of a Choice Set in the Study

The number of attributes in this study was eight. Therefore, we had to determine the feasibility of using a CBC despite the large number of attributes, or if the larger effort of drawing up an ACBC would be needed. The form of the attributes provided an important aspect in making this decision. The amount of information that a test subject has to absorb and process in connection with every single attribute is especially important when calculating the reasonable maximum number of attributes. If the attributes being examined are graphic elements (e.g., app icon) or information, which can be quickly understood (e.g., price), then CBC could be a feasible option to carry out this type of analysis with more than six attributes [29].

Due to these criteria and considering the impact of an ACBC on the interview duration, CBC appeared to be the more suitable choice for the planned empirical survey. As far as survey design was concerned, it was important to define the form of the stimuli, specifically the question of which combination of attribute variations would constitute the stimuli and how the stimuli should be presented to each test subject. Here, the Full Profile Method was used, in which each product profile consists of all the attributes. As the number of attributes

was already very high, we decided to present only two stimuli at a time so as not to overstrain the test subjects with regard to the information they had to evaluate. In order to create a selection situation as close as possible to a real-life purchase situation, a “none option” was also included.

Figure 2 shows a complete selection situation as an example of how it also appeared in the final survey. In addition to the (randomly) created selection sets, so-called hold-out sets were integrated into the survey. These special selection sets serve to analyze the validity of the prognosis. They are not integrated into the benefit evaluation and are used to evaluate the quality of the prognosis of the preference rating. Two of these sets were defined and included.

The conjoint analysis was carried out using the *Sawtooth SSI Web 7* software package [30]. The main objective of the study was to measure the importance of the presented app store elements for mobile application purchase decisions. The study was conducted as an online survey. The website for the online survey was generated by the *SSI Web 7* software, based on the aforementioned study design. The configuration of the CBC analysis and selected configuration parameters are summarized in Table I.

TABLE I. CONFIGURATION OF THE CBC ANALYSIS

Parameter	Value
Number of Random Choice Tasks	12
Number of Fixed Choice Tasks	2
Number of Concepts per Choice Task	2 (and an additional “none option”)
Response Type	Discrete Choice (single select radio button)
Advanced Design Module Settings	Traditional Full-Profile CBC Design
Randomize Attribute Position within Concepts	No Randomize of Attribute Order

B. Latent Class Analysis

The idea behind consumer market segmentation is to divide the market into smaller homogenous groups for the purpose of product placement and targeted marketing [31]. By doing so, it becomes possible to better adjust the product and marketing efforts to consumer preferences or user requirements. According to [32], two approaches to market segmentation are a priori, aka common sense, or post hoc (i.e., data driven). A priori segmentation would define segments based on obvious group characteristics such as age, gender, geographical region and other general demographic information (e.g., men over 50 years living in a specific area). As a priori segmentation needs no analysis, it is much easier to select homogenous groups. While this approach might already be more effective than mass marketing, it relies on the discriminating power of directly observable group characteristics and ignores underlying variations of product needs and preferences of the

individual user or consumer. Accordingly, post hoc segmentation tries to look at the results of studies specifically designed to understand the potential user's needs and preferences.

Conjoint based preference data can be used for segmentation based on latent class analysis (LCA) [33][34]. Having gained popularity in the 1990's, the model "detects segments of respondents having similar preferences based on their choices in CBC questionnaires" [35]. Latent class analysis takes CBC one step further in that it identifies groups of respondents that share specific preferences and estimates the average part-worth utility for each of the groups of respondents. In other words, the approach can be used to "discover segments of respondents who tend to have similar preferences manifest within the CBC (choice-based conjoint) data" [36].

In an LCA, the segmentation process is initiated by randomly selecting estimates of each group's part-worth utility values and then estimating the probability that a given respondent belongs to a specific group. Summing the logs of those probabilities, for all respondents across all questions results in the log-likelihood. In an iterative approach those probabilities are used to recalculate the logit weights until a "convergence limit" is reached [37]. Solutions can be calculated for a different numbers of groups. To determine the best number of groups, the log-likelihood cannot be used as it typically moves closer to zero as the number of segments increases. Thus, goodness-of-fit or Information Criteria (ICs) are used to determine the number of segments.

Two of the most common ICs are Akaike's Information Criterion (AIC) and the Bayesian Information Criterion (BIC) [38]. Often these ICs are automatically calculated by statistical packages, e.g., the Sawtooth software used in this study, with the expectation that they will provide some guidance with regards to selecting the appropriate number of segments. That being said, from a managerial perspective "the most important aspects to consider when choosing a solution for segmentation purposes are its interpretability and stability (reproducibility)" [36]. Once the group size has been determined, the data can be interpreted using the segment specific attribute importance as well as part-worth utilities rescaled for comparability.

V. STUDY FINDINGS

Based on our methodical considerations an empirical study was conducted. The study was based on the presentation of the app store elements as discussed in the previous section in an online questionnaire. The survey was online between December 19, 2013 and January 10, 2014. Participants were acquired by using social media and various other online and offline channels of the RheinMain University of Applied Science in Wiesbaden, Germany. A total of 221 people participated in the conjoint analysis interview. Of these, 163 completed the interview in its entirety and are, therefore, included in the subsequent evaluation. Selected demographic characteristics of the study participants are shown in Table II below. The demographics show that the study might be biased by the participating media and design students and due to the resulting high proportion of iOS users compared to the lower usage rate in the total population in Germany of around 32 percent

at the end of 2013 [39] and the underrepresentation of older user segments.

TABLE II. DEMOGRAPHICS OF THE STUDY PARTICIPANTS

Characteristics	Absolute Number	Percentage
Mobile OS		
Apple iOS	78	47.9%
Android	78	47.9%
Blackberry OS	1	0.6%
Windows Phone/Mobile	5	3.1%
Symbian	1	0.6%
Purchased Apps		
None	32	19.6%
1-5	40	24.5%
6-10	20	12.3%
11-20	19	11.7%
21+	52	31.9%
Gender		
Female	70	42.9%
Male	93	57.1%
Ages		
18-24	66	40.5%
25-34	70	42.9%
35-44	21	12.9%
45-54	5	3.1%
55+	1	0.6%

The evaluation of the collected data took place in two steps: In the first phase, a counting analysis was conducted. This analysis can be used to calculate an outline of so called main effects. A main effect of an attribute level is calculated here as a proportion and reveals how many times a specific attribute level was chosen, divided by the number of times this attribute level was available for choice in the testing. Counting analysis is a simple way to get a first indication of the relevance of the attribute levels. As a second step, the part-worth utilities of the attribute levels were estimated based on a logit analysis to find the maximum likelihood solution for the data. Based on the results of the part-worth utility estimation, the relative importance of the individual app store elements were finally determined.

A. Counting Analysis

A counting analysis and the proportions that are calculated at this stage can be used to identify the "winner" of the different attribute levels. Table III shows the results of the counting analysis for all attributes and attribute levels considered in this study. The higher the proportion of an attribute level is, the stronger this attribute level may have influenced the choice of participants. For the app store element "Reviews (stars)" a five-star rating was the "winner" – which is not surprising. However, in comparison, choices with this attribute level were selected more than twice as often (0.421/0.158) as choices with no stars in the reviews.

TABLE III. SUMMARY OF STUDY RESULTS

Attributes and Attribute Levels	Counts (Proportions of "Wins")	Part-Worth Utilities
App Icon		
High quality	0.312	0.22215
Medium quality	0.262	-0.01639
Low quality	0.234	-0.20575
App Name		
SafeTalk – Your safe messenger	0.277	0.02744
SafeTalk Secure Messenger	0.247	-0.10392
Safetalk with AES-256 Encryption	0.283	0.07648
Reviews (stars)		
5 stars	0.421	0.73209
3 stars	0.229	-0.13465
No stars	0.158	-0.59744
Number of Reviews		
7.240 reviews	0.329	0.31666
310 reviews	0.320	0.26487
5 reviews	0.229	-0.19484
No reviews yet	0.198	-0.38669
Price		
Free of charge	0.385	0.60605
0.89 EUR	0.274	0.02966
1.79 EUR	0.238	-0.14028
2.69 EUR	0.180	-0.49543
Screenshots		
High quality	0.262	-0.02198
Medium quality	0.274	0.01437
Low quality	0.271	0.00760
App Description		
High quality	0.283	0.07434
Medium quality	0.269	0.01472
Low quality	0.256	-0.08906
Server Location		
Germany	0.373	0.52316
USA	0.224	-0.20529
Unknown	0.212	-0.31788

However, as mentioned before, this analysis can give a first indication of the relevance but does not provide measurements for the part-worth utilities of attribute levels and relative importance of the different attributes, i.e., app store elements.

B. Estimation of Part-worth Utilities

Part-worth utilities were calculated by using the multinomial logit estimation provided by the *Sawtooth* software for the CBC analysis. For the model estimation, a Chi Square of 473.7 was reported. Considering 18 degrees of freedom (26 attribute levels and 8 attributes) the Chi Square is much larger than the required 34.8 for a 0.01 level, which would mean that the choices of the respondents are significantly affected by the attribute composition [27]. The estimated part-worth utilities

represent the relative desirability of an attribute level. The higher the value of a part-worth, the greater the impact of the corresponding attribute level on the buying decision. Part-values are automatically standardized, so that the result per attribute amounts to „0“. Reciprocally, this means that negative values can also arise. Table III shows the estimated values for all attribute levels. These should be interpreted to mean that a higher number corresponds to a higher part-worth utility and that this attribute variation therefore had a higher preference among the test subjects. If we look again at the attribute “Reviews (stars)“, it becomes evident that the attribute level „5 stars“ has a very high part-worth value with a positive value of 0.73209. The other two variations „3 stars“(-0.13465) and „no stars“(-0.59744) were less important for the purchase decision of the test subjects due to smaller values of the corresponding part-worth utilities.

C. Calculation of the Attribute Importance

The defined objective of the empirical study was not only to find out the utilities of the attribute variations but also to analyze each individual app store element in terms of its relative importance for an app purchase decision. Therefore, we must find a unit of measurement to express the relative importance of each attribute. The calculation is carried out by dividing the range of the part-worth of each attribute by the sum of the part-worth ranges of all the attributes. Hereby, the range is defined as the difference between the highest and the lowest part-value within the levels of an attribute [40]. The results can be seen in Table IV.

TABLE IV. RELATIVE IMPORTANCE OF ATTRIBUTES

Attribute	Attribute Importance
Reviews (Stars)	27.8%
Price	23.2%
Server Location	17.6%
Number of Reviews	14.9%
App Icon	9.0%
App Name	3.6%
App Description	3.3%
Screenshots	0.6%
Total	100.0%

The values reveal that the reviews according to the star principle have the largest influence on the purchase decision. Almost 28% of the decisions are based on this criterion. The highest part-worth utility and/or the most positive influence was of course an app review with 5 stars. The distance to the other attribute variations (3 stars, no stars) was the highest with this app store element compared to the other elements. This highlights the extremely high relevance of good reviews and the importance of this attribute for the perceived total utility of the corresponding app presented in the app store. As was to be expected, pricing has a high level of importance for the purchase decision, too. The test subjects reacted in a very price-sensitive way. It should also be noted that many apps are now offered at the Apple App Store for free or at a greatly reduced price at the beginning or at some stage of their life

cycle for a certain period of time. A certain „freebie“ mentality is also reflected in the order of precedence in this study and shows that price is one of the most important criteria for an app. The app provider’s server location differs from the other elements in as far as it is not a standardized app store element but the app developer’s company-related element. Therefore, we can conclude that users not only include the app store’s design elements into their purchase decisions, but also consider and evaluate outstanding and specific properties of the app. In this case, there was a particularly positive effect on the purchase decision if the messenger provider was located in Germany. The number of reviews relates to the reviews according to the star principle. Here, we see the tendency that the part-worth utility is perceived as higher, the more reviews an app has. An interesting aspect here is that the part-worth of the extreme scenario considered in the survey with 7,240 reviews did not substantially differ from the next level with 310 reviews. The distance to the next two steps (5 reviews, no reviews) is considerably larger, however. This means that an optimal number of reviews – which can be attained with a reasonable amount of effort on the part of the app provider – can be assumed to be more than 5, but not significantly higher than 310 reviews. The app icon is considerably less important than expected. Besides the screenshots and the star reviews, it is the third graphic element and easy for the potential buyer to understand. Nevertheless, the test subjects apparently did not assess the quality of the app on the basis of the icon but stuck to the very much more rational criterion of the reviews when making their purchase decision. The app name is of very low significance. Many users see it as a “frill” within the overall impression of the app store and it is therefore of little interest. The study results even show that the name “Safetalk with AES-256 Encryption,” which was previously defined as the worst variation, actually had the highest partial benefit value. However, this could be a result of the specific setup and the sensitivity of the app users towards data security in Germany. The complicated name – even if not understood by the customers – may be associated with a highly sophisticated technological solution to protect the user from the danger of interception. The app’s descriptive text is also of little importance in terms of decision making. This suggests that potential buyers do not take the time to read it or may be very familiar with the type of apps that have been tested here. It should be noted at this point that the descriptive texts used in the survey were relatively short. In real life, an app is mostly described in much more detail and using many more characters – the attention span could, therefore, be even shorter than for the texts used in the survey.

With a relative importance of 0.6 percent, the screenshots had the lowest influence on the purchase decision. Here, too, it was striking that the part-worth of the medium quality screenshots was the highest, followed by those of the worst quality. The highest quality level had the lowest part-worth value for the test subjects. Here we should note, however, that the differences recorded were marginal and the general result, i.e., that screenshots hardly influence purchase decisions, is predominant. This may also be due to the fact that the subject of the study, messaging app functions, is relatively well-

known and simple and that therefore screenshots have only minor informational value as far as the app is concerned.

D. Group Segmentation

In order to better understand if these preferences are universal or if the user preferences fall into different groupings according to common preferences, a LCA was conducted as discussed in Section IV. Solutions were computed with the Sawtooth software package for a minimum of two and a maximum of seven groups considering typical ranges used in LCA studies [41][42]. As shown in Table V, the log-likelihood moves closer to zero as more segments are included in the solution. To determine the number of groups, the ICs mentioned in the methodology needed to be analyzed. The most common used ICs in LCA studies are Akaike’s Information Criterion (AIC) and Bayesian Information Criterion (BIC) as mentioned before. Both criteria are based on the likelihood function but incorporate penalties to control for over fitting (to derive a parsimonious solution). The goal is to minimize the value of the IC where the lowest value indicates the best fitting model.

TABLE V. INVORMATION CRITERIA OF THE LCA

Group	Log-likelihood	AIC	BIC
2	-1637	3351	3569
3	-1562	3243	3572
4	-1497	3152	3593
5	-1437	3071	3624
6	-1409	3056	3720
7	-1373	3023	3799

However, as shown in the Table V, the two IC produce contrary results by supporting the two (BIC) or the seven (AIC) group solution. Such ambiguous results are not unusual in LCA and so [38] suggest that the choice of an IC has to consider the goal of the study. In this context, the BIC-preferred size can be interpreted as the minimum size for a parsimonious model and the AIC-pref erred size as a maximum when the exploration of population heterogeneity is in focus. The choice then has to be made “based on other kinds of fit criteria, on theory, or on subjective inspection of results” [38]. As the aim of the study at hand was to explore the population heterogeneity, a subjective inspection of the two-group solution offered only limited insights into potential market segments. On the other hand, the seven-group solution was selected for further interpretation and provided clear groupings with distinct preferences.

As discussed before in the methodology section, the relative importance of the attributes and the preferred attribute levels can be used to interpret the preference structures of the computed user segments. In a first step, the group specific relative importance of the attributes was inspected. Considering an equal relative importance of each of the eight attributes for the group members, a relative importance of 12.5 percent could be expected. Accordingly, those attributes with a relative importance greater than 12.5 percent could be interpreted as truly impacting the purchase decision. In Tables VI through XII, those attributes with a relative importance greater than

12.5 percent were highlighted in grey. Once the attributes of importance were identified, the preferred attribute level was determined based on the group specific estimation of the rescaled part-worth utilities. At the attribute level, the (rescaled) part-worth utilities for each of the three versions of the app element were compared. Accordingly, the version with the highest part-worth utility represented the preferred configuration of that app store element. The information about the group specific importance of the attributes and the preferred group levels were then used to define an appropriate characterization of preference structure for the presentation for app store elements.

The first group shown in Table VI was focused on quality as indicated by the high preference for 5 star ratings. They were also very interested in the server location being in Germany. Price was not as important and the group members would prefer to pay a moderate price (1.79 EUR) for an app with the appropriate quality features. From this perspective the mindset of this group can be characterized as “Quality for Money”. More than 13 percent of the sample (N=23) have been assigned to this segment.

TABLE VI. LCA MARKET SEGMENTATION: GROUP 1

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 1 (“Quality for Money”, N=23)		
Server Location	44%	Germany
Reviews (stars)	16%	5 stars
App Description	12%	High quality (User-oriented)
App Icon	11%	Medium quality (Balanced)
App Name	6%	Safetalk w. AES-256 Encryption
Number of Reviews	6%	7.240 reviews
Screenshots	3%	High quality (Notated)
Price	2%	1.79 EUR

Similar to the “Quality for Money” group, the preferences in the next group, presented in Table VII, were heavily influenced by quality as indicated by the importance of 5 star ratings, the server location in Germany, and the preferred high number of user reviews.

TABLE VII. LCA MARKET SEGMENTATION: GROUP 2

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 2 (“Free Rider”, N=30)		
Reviews (stars)	23%	5 stars
Server Location	20%	Germany
Price	19%	Free of charge
Number of Reviews	18%	7.240 reviews
App Description	7%	Medium quality (Tech Savvy)
App Name	5%	SafeTalk – Your safe messenger
App Icon	5%	High quality (Modern/Specific)
Screenshots	5%	Med. quality (Design-focused)

The main differentiating factor being that they did not want to pay for it. Given that they expect high quality for free, this segment was characterized as the “Free Rider” segment. More than 18 percent (N=30) of the respondents are classified in this group.

Just the opposite can be seen in group three. As seen in Table VIII, this group is actually willing to pay “top dollar” for high quality apps. Accordingly, this group has been described as the “Premium” group.

TABLE VIII. LCA MARKET SEGMENTATION: GROUP 3

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 3 (“Premium”, N=14)		
Reviews (stars)	22%	5 stars
Price	20%	2.69 EUR
App Icon	14%	High quality (Modern/Specific)
Server Location	11%	Germany
App Name	11%	Safetalk w. AES-256 Encryption
App Description	10%	Medium quality (Tech Savvy)
Number of Reviews	9%	310 reviews
Screenshots	3%	Med. quality (Design-focused)

In this case, quality is seen as 5 star ratings and usage of the modern/specific design icon. This segment is on the smaller side, accounting for only 9 percent (N=15) of the respondents, but their willingness to pay a high price for an application make them a relevant segment.

Similar in size is the “free at all costs” group seen in Table IX. This group has been deemed so as price is the only attribute that matters to them.

TABLE IX. LCA MARKET SEGMENTATION: GROUP 4

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 4 (“Free at all Costs”, N=15)		
Price	67%	Free of charge
Reviews (stars)	8%	5 stars
Screenshots	7%	Low quality (Functional)
App Name	5%	Safetalk w. AES-256 Encryption
Number of Reviews	4%	310 reviews
Server Location	3%	USA
App Description	3%	Medium quality (Tech Savvy)
App Icon	3%	Medium quality (Balanced)

This group also accounts for about 9 percent (N=15) of the respondents but unlike the premium group, if the app is not free, this group will most likely not consider it. Similarly, the largest segment, accounting for 33 percent (N=54) of respondents, also prefers apps that are free of charge, however, that is less important than high ratings and reviews. This group has been labeled the “Socially Motivated Majority” group.

TABLE X. LCA MARKET SEGMENTATION: GROUP 5

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 5 ("Socially Motivated Majority", N=54)		
Reviews (stars)	30%	5 stars
Number of Reviews	25%	7.240 reviews
Price	20%	Free of charge
App Icon	8%	High quality (Modern/Specific)
Server Location	7%	Germany
App Description	6%	High quality (User-oriented)
App Name	4%	SafeTalk – Your safe messenger
Screenshots	1%	Low quality (Functional)

As seen in Table X, the members of this group are heavily influenced by the experiences and ratings of other users and thus might be inclined to follow "word-of-mouth". Confirming what was assumed during the design of the app elements, group 6 is a perfect example of the German customer's heightened awareness about data security, accounts for 13% of the respondents this group is very concerned with security and has thus received the title "privacy concerned"

TABLE XI. LCA MARKET SEGMENTATION: GROUP 6

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 6 ("Privacy Concerned", N=22)		
Server Location	52%	Germany
App Icon	12%	High quality (Modern/Specific)
Number of Reviews	10%	310 reviews
Reviews (stars)	9%	5 stars
App Description	8%	Low quality (Complex)
Price	6%	Free of charge
App Name	2%	SafeTalk Secure Messenger
Screenshots	2%	Low quality (Functional)

As seen in Table XI, server location is of utmost importance, so much so that it is the only attribute influencing their purchase decision. Most likely due to the recent NSA disclosures [22], they have a strong preference for a server location in Germany. However, their willingness to pay for this is questionable. Table XII shows the outlier group. Accounting for only 3% of the respondents, this group is not only small, their preferences did not conform to any expectations.

TABLE XII. LCA MARKET SEGMENTATION: GROUP 7

Groups and Attributes	Relative Importance	Preferred Attribute Level (Part-worth Utilities)
Group 7 (Outliers/Not Considered, N=5)		
Price	34%	Free of charge
Number of Reviews	20%	No reviews yet
Reviews (stars)	12%	5 stars
App Name	8%	SafeTalk – Your safe messenger
Screenshots	7%	Low quality (Functional)
App Description	7%	Low quality (Complex)
App Icon	6%	Medium quality (Balanced)
Server Location	6%	Unknown

Accordingly, this group has not been analyzed to any further extent. However, the existence of such a group indicates that the decision to use a maximum of 7 groups the LCA was a good estimate.

VI. IMPLICATIONS

This study confirms the observation from best practices, that reviews have a major influence on the user's purchase decision. Not only from a mass market perspective, but also among the majority of the market segments. Average ratings according to the star principle as well as the number of reviews given determine the buying decision of an app to a very large degree. These two criteria, however, cannot be directly influenced by the app provider – reviews are made by the app user and are published by the app store with no prior screening. Nevertheless, there are numerous possibilities for the provider to influence the reviews, at least to some extent. Active review management should therefore be conducted. Review reminders within the app can for example help to continuously increase the number of reviews. It is advisable to wait for a certain period of time before displaying review reminders as the probability of receiving a positive review is higher when the app has been used for a period of time. Reviews can also be stimulated by actively reacting to user feedback, i.e., by responding to reported software bugs or considering suggestions for improvements in upcoming updates.

The possibilities for the provider to influence the price are often strongly determined by the costs. In addition, the price decision can depend on the app's life cycle or even some important seasonal factors (special offers on public holidays for example). Thus, a low price level may not be an option and the findings of the conjoint analysis cannot be transferred to a general recommendation on an adequate pricing strategy. However, if it makes sense for the type of app in question, a free version can be offered, which can be supplemented by additional content per in-app-purchase. This "freemium model" takes the user's initial price-sensitivity into account. Revenue generation is then postponed to a later phase of usage. Alternatively, the results of the Latent Class analysis suggest that there are certain groups willing to pay a premium for quality applications. Accordingly, while mass marketing may be less effective for higher priced apps, effectively targeting the appropriate market segment can be another solution in such situations.

Another important finding is that particular attention should be drawn to app-specific properties if these could positively influence sales. From a mass market perspective, this applied to the server location of the company providing the app and the corresponding messenger service. In this particular case, it appears to have addressed a basic need for security among the test subjects. This may not be directly transferable to other apps. However, such "unique selling propositions" should be particularly highlighted and communicated via the other elements. This is especially important from a market segmentation perspective because for one of the groups, the location of the server was the main influencer in the ultimate purchase decision.

The elements not yet mentioned at this point (app icon, app name, descriptive text, screenshots) should by no means be neglected during the course of marketing activities. From a mass market perspective these have a smaller overall influence on the customer's purchase decision and only have a limited ability to set the product apart from the competition. That being said, the segmented markets show that these elements have a greater influence over some of the groups. Accordingly, such elements must indeed be well designed, in order to convince a customer to purchase or to use the app. This is especially true when marketing to desired market segments. The descriptive text and the app name, for example, are nevertheless crucial for the app store's search algorithms to enable the mobile application to be found at all. Whether the app name is easy to remember is another factor that plays an important role in the selection process and in word-of-mouth propaganda.

Furthermore, it can be expected that in a perfect world, a killer app would be created and launched in the app store. As soon as it is launched, all potential users would be exposed to it and have the opportunity to download it and thus provide ratings which will promote further usage. However, generally speaking the app market is not a "field of dreams" and just because an app is built it does not mean "they" will come. As originally expected, there are different homogenous segments with very specific preferences with regards to the various app elements. Accordingly, effectively targeting the appropriate market segment can not only increase awareness and potential downloads, but also increase the chance that the app meets the user's needs and interests, which will result in higher ratings. As discussed earlier, outside of placing an app in the appropriate category, app stores do not offer the ability, via their platform, actively market to specific market segments. Thus, this study shows the need for such a tool as developers and marketers looking to reach these segments are currently forced to use external channels.

VII. CONCLUSIONS

This study has revealed some empirically based recommendations on how to align the elements of the app presentation in app stores to customer preferences. The results of the CBC analysis showed four main attributes of importance. While developers and marketers can influence two of the attributes the other two are a result of feedback from other users. While these four elements had the greatest over all influence over a user's ultimate purchase decision, the results of the LC analysis showed that different user segments have very different preferences and needs. While one group was very heavily influenced by price another was influenced by server location. This indicates that while the overall trend is to offer applications that are free (not counting for in app purchase options) there are still those that are willing to pay a premium for quality. Accordingly, appropriate market segmentation can help developers better reach their intended market and stand out in the vast sea of applications.

The findings, however, refer to a rather small and not representative sample. Moreover, the generalizability of the study is limited due to the fact that here just one single, specific application was investigated, using the example of select design

elements of the Apple App Store. More detailed studies in different application domains and with regard to different app stores will be necessary in order to verify the validity of the findings derived in this study.

REFERENCES

- [1] S. Böhm and S. Schreiber, "Mobile App Marketing: A Conjoint-based Analysis on the Importance of App Store Elements," CENTRIC 2014, The Seventh International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2014, pp. 7–14.
- [2] J. F. Hughes, *iPhone and iPad apps marketing. Secrets to selling your iPhone and iPad apps*. Que Pub., Indianapolis, 2012.
- [3] J. Mayerhofer, *Apps erfolgreich verkaufen. Vermarktungsstrategien für Apps auf iPhone, iPad, Android und Co*. Hanser, München, 2012.
- [4] P. Kotler and G. Armstrong, *Principles of marketing*. Pearson, Upper Saddle River, N.J, 2014.
- [5] D. Wooldridge and M. Schneider, *The business of iPhone and iPad App Development. Making and marketing Apps that succeed*. Apress; Springer Science+Business Media, LLC, New York, 2011.
- [6] R. Mroz, *App-Marketing für iPhone und Android. Planung, Konzeption, Vermarktung von Apps im Mobile Business*. mitp, Heidelberg, 2013.
- [7] F. Hamka, H. Bouwman, M. de Reuver, and M. Kroesen, "Mobile customer segmentation based on smartphone measurement," *Telematics and Informatics* 31, 2, 2014, pp. 220–227.
- [8] N. Kuh, Ed., *Foundation iPhone App Development*. Apress, Berkeley, CA, 2012.
- [9] L. Jordan, Ed., *Beginning iOS 5 Games Development*. Apress, Berkeley, CA, 2011.
- [10] E. Kim, *The best book on marketing your android app*. CreateSpace Independent Publishing.
- [11] M. Amerson, *The best book on IOS app marketing*. Hyperlink Press, 2012.
- [12] S.-Y. Ihm, W.-K. Loh, and Y.-H. Park, "App Analytic: A Study on Correlation Analysis of App Ranking Data," 2013 International Conference on Cloud and Green Computing (CGC), pp. 561–563.
- [13] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Ranking fraud detection for mobile apps," *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pp. 619–628.
- [14] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Discovery of Ranking Fraud for Mobile Apps," *IEEE Trans. Knowl. Data Eng.* 27, 1, 2014, pp. 74–87.
- [15] B. Ifrach and R. Johari, "Pricing a bestseller," *SIGMETRICS Perform. Eval. Rev.* 41, 4, 2014, p. 51.
- [16] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," 2013 10th Working Conference on Mining Software Repositories (MSR). *Proceedings*, May 18–19, 2013, San Francisco, CA, USA, pp. 41–44.
- [17] P. Yin, P. Luo, W.-C. Lee, and M. Wang, "App recommendation," *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013*, pp. 395–404.
- [18] J. Muller and M. Lillack, "Conjoint Analysis of Software Product Lines: A Feature Based Approach," 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), pp. 374–377.

- [19] D. Eckjans, J. Fahling, J. M. Leimeister, and H. Krcmar, "Mobile Customer Integration: A Smartphone Application Prototype for Conducting Mobile Conjoint Studies," 2011 Tenth International Conference on Mobile Business, ICMB, pp. 129–135.
- [20] R. Sandberg and M. Rollins, *The business of Android Apps development. Making and marketing Apps that succeed on Google Play, Amazon App Store and more.* Apress, New York, 2013.
- [21] M. Flarup, *iPhone App Icon Design: Best Practices.* <http://www.pixelresort.com/blog/iphone-app-icon-design-best-practises/> 2014.08.01.
- [22] H. Schmundt and G. Traufetter, 2014, *Digital Independence: NSA Scandal Boosts German Tech Industry.* <http://www.spiegel.de/international/business/german-it-industry-looks-for-boom-from-snowden-revelations-a-950786.html> 2014.08.12.
- [23] R. Luce and J. W. Tukey, "Simultaneous conjoint measurement: A new type of fundamental measurement," *Journal of Mathematical Psychology* 1, 1, 1964, pp. 1–27.
- [24] C. Breidert, *Estimation of Willingness-to-Pay. Theory, Measurement, Application.* DUV, Wiesbaden, 2006.
- [25] A. Gustafsson, A. Herrmann, and F. Huber, 2007, "Conjoint Analysis as an Instrument of Market Research Practice," *Conjoint measurement. Methods and applications.* A. Gustafsson, A. Herrmann and F. Huber, Eds. Springer, Berlin, New York, pp. 3–30.
- [26] P. E. Green and V. Srinivasan, "Conjoint analysis in marketing: New developments with implications for research and practice," *Journal of Marketing* 54, 4, 1990, pp. 3–19.
- [27] Sawtooth Software, 2013, *The CBC System for Choice-Based Conjoint Analysis.* Technical Paper Series. <https://sawtoothsoftware.com/download/techpap/cbctech.pdf> 2014.08.01.
- [28] P. E. Green, A. M. Krieger, and Y. Wind, "Thirty Years of Conjoint Analysis: Reflections and Prospects," *Interfaces* 31, 3, 2001, pp. S56-S73.
- [29] Sawtooth Software, 2014, *The Adaptive Choice-Based Conjoint (ACBC) Technical Paper.* <http://www.sawtoothsoftware.com/download/techpap/acbctech.pdf> 2014.08.01.
- [30] Sawtooth Software, 2014, *What is SSI Web?* <http://www.sawtoothsoftware.com/support/downloads/download-ssi-web/what-is-ssi-web> 2014.09.19.
- [31] N. Djokic, S. Salai, R. Kovac-Znidarsic', I. Djokic, and G. Tomic, "The Use of Conjoint and Cluster Analysis For Preference-Based Market Segmentation," *Inzinerine Ekonomika-Engineering Economics* 24, 4, 2013, pp. 343–355.
- [32] S. Dolnicar and F. Leisch, "Using graphical statistics to better understand market segmentation solutions," *Int. J. Market Res.* 56, 2, 2014, pp. 207–230.
- [33] W. S. Desarbo, V. Ramaswamy, and S. H. Cohen, "Market Segmentation with Choice-Based Conjoint Analysis," *Marketing Letters* 6, 2, 1995, pp. 137–147.
- [34] W. S. Desarbo, M. Wedel, M. Vriens, and V. Ramaswamy, "Latent Class Metric Conjoint Analysis," *Marketing Letters* 3, 3, 1992, pp. 273–288.
- [35] Sawtooth Software, Inc, *Latent Class v4.5*, 2012.
- [36] Sawtooth Software, 2004, *The CBC Latent Class Technical Paper (Version 3).*
- [37] Sawtooth Software, 2004, *The CBC Latent Class Technical Paper (Version 3).*
- [38] J. J. Dziak, D. L. Coffman, S. T. Lanza, and R. Li, *Sensitivity and specificity of information criteria.* Technical Report Series 12-119. The Methodology Center, The Pennsylvania State University.
- [39] StatCounter, 2014, *Top 8 Mobile Operating Systems in Germany from July 2013 to July 2014.* http://gs.statcounter.com/#mobile_os-DE-monthly-201307-201407 2014.08.01.
- [40] B. K. Orme, *Getting started with conjoint analysis. Strategies for product design and pricing research.* Research Publishers, LLC, Madison, WI, 2006.
- [41] C. Chan-Halbrendt, E. Zhllimab, G. Sisiorc, and D. Imamid, "Consumer Preferences for Olive Oil in Tirana, Albania," *International Food and Agribusiness Management Review* 13, 3, 2010, pp. 55–74.
- [42] E. Zhllima, C. Chan-Halbrendt, Q. Zhang, D. Imami, R. Long, L. Leonetti, and M. Canavari, "Latent Class Analysis of Consumer Preferences for Wine in Tirana, Albania," *Journal of International Food & Agribusiness Marketing* 24, 4, 2012, pp. 321–338.

A Network-disaster Recovery System using Multiple-backup Operation Planes

Toshiaki Suzuki, Hideki Endo, Isao Shimokawa,
and Kenichi Sakamoto

Research & Development Group
Hitachi, Ltd.
Kanagawa, Japan

{toshiaki.suzuki.cs, hideki.endo.es, isao.shimokawa.sd,
and kenichi.sakamoto.xj}@hitachi.com

Hidegori Inouchi, Taro Ogawa, Takanori Kato,
and Akihiko Takase

Information & Telecommunication Systems Company
Hitachi, Ltd.
Kanagawa, Japan

{hidenori.inouchi.dw, taro.ogawa.tg, takanori.kato.bq,
and akihiko.takase.wa}@hitachi.com

Abstract—A “network-disaster recovery system” using multiple-backup operation planes is proposed. Under this system, a whole network is separated into multiple areas. Before starting network operations, a network-management server calculates recovery paths for every possible failure in network area and distributes them with a recovery identifier (ID) for each network-area-failure pattern (on the backup operation plane). Network nodes receive and store the recovery IDs and recovery configurations. The network-management server determines a failure pattern after detecting the network-area failures and distributes the recovery ID to related network nodes. The network nodes that received the recovery ID start data transmission according to the path configurations specified by the recovery ID. After the completion of these procedures, the network-area failures are swiftly recovered. A prototype system (composed of a network-management server and 96 simulated packet-transport nodes) with a graphical viewer was implemented, and its performance was evaluated. According to the results of the evaluation, all recovery-path configurations for 1000 pseudo-wires (PWs) (namely, transmitting the recovery ID to the related network nodes and using a recovery-path database specified by the ID) were done within 100 milliseconds after the network-area failures were detected. On the condition that the configuration time depends on the size of the recovery-path database, the proposed system takes about one minute and 40 seconds in the case of 1,000,000 PWs. On the other hand, a restoration scheme under the same evaluation conditions used for the proposed system takes over 10 minutes to recalculate recovery paths from detection of the first area-based network failure. That is, the proposed recovery scheme can recover network-area failures faster than the conventional restoration scheme can.

Keywords - *network management; disaster recovery; packet transport; reliable network*

I. INTRODUCTION

Lately, as reflected in the rising number of Internet users and the popularity of cloud services, applications and services provided by way of networks have become indispensable in daily life. Network services must, therefore, be highly reliable and “always available”. When extensive disasters occur, network services could be out of service for a long time. Consequently, networks must be robust enough so that they can continue to provide services even if their facilities are extensively damaged. In our previous study, presented at INNOV 2014 [1], entire system architecture was

focused on. In this extended work, a prototype system was implemented, and its performance was evaluated in comparison with a conventional system.

As recovery procedures for network failures, two major techniques [2] are applied: “protection,” by which recovery paths are physically prepared in advance of network failures by allocating extra network resources; and “restoration,” by which recovery paths are “calculated” after network failures are detected.

Protection is easily applied to multi-layer networks, and recovery is immediate because recovery paths are prepared in advance (that is, before network operations are started). However, if the prepared recovery paths are not available when network failures occur, network-connection services will become out of service. On the other hand, if restoration is applied, network connections can be recovered if recovery paths are recalculated after network failures are detected. However, it takes more time to recalculate the recovery paths if the operated networks are huge and have many network nodes. Therefore, if huge quantities of paths are used to transmit data packets, much time is needed to recalculate all recovery paths, and the network will not recover from a disaster expeditiously. In addition, even if network connections are recovered, all network flows will try to use the same recovery path. As a result, the network will easily become congested, making it difficult to guarantee network-transmission quality.

In light of the above-described issues, a robust network-management scheme is required. Specifically, it must control multi-layer-network resources so as to provide and maintain network-connection services at times of a “network disaster” (namely, a catastrophic failure of a network). To achieve that control, a network-management system has to monitor and control the multi-layer-network resources.

The overall aim of the present study is to develop a network-management scheme for monitoring and controlling multi-layer network resources so as to provide robust networks that can swiftly recover from a network disaster. To swiftly recovery from a network disaster, three steps should be followed: the first step is to find network failures in a short time; the second is to promptly determine how to recover the network; and the third is to immediately configure recovery paths. In the present study, the second step is focused on, and a “network-disaster recovery system”

using an area-based network-management scheme, which controls networks composed of IP networks and packet-transport networks, such as the Multi Protocol Label Switching - Transport Profile (MPLS-TP) network, is proposed.

The rest of this paper is organized as follows: Section II describes related work. Section III explains the requirements concerning a network-disaster recovery system. Section IV proposes a network-disaster recovery system that meets those requirements. Sections V and VI respectively describe an implementation of a prototype system and present some results of evaluations of the system's performance. Section VII concludes the paper.

II. RELATED WORK

Regarding highly available and reliable network management, several standardization activities have been ongoing. For example, MPLS-TP-related operation, administration, and maintenance (OAM) functions have been standardized. In the first stage of that standardization, the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) [3] discussed specifications such as Transport - Multi Protocol Label Switching (T-MPLS). In the next stage, the ITU-T jointly standardized MPLS-TP specifications with the Internet Engineering Task Force (IETF) [4]. A request for comments (RFC) on requirements of MPLS-TP [5] was issued as the first step. A framework of MPLS-TP was documented as RFC 5921 [6]. Using MPLS-TP OAM [7] functions makes it easier to detect failures in transport networks. In addition, an RFC on a framework for MPLS-TP survivability [8] was issued. In relation to the proposed system, it is useful to detect network failures promptly in order to determine areas that are out-of-service.

With regards to failure recovery, two major techniques, namely, "protection" and "restoration," have been proposed. By means of protection, a standby path is preliminarily calculated and established by using extra physical resources. When network failures are detected, an active path is promptly changed from the current path to the standby path.

One of major recovery schemes, called "fast reroute" [9], prepares a back-up path. In addition, a recovery scheme combining an IP layer and an optical layer was proposed [10]. A scheme for preparing multiple backup paths to tackle multiple failures was presented [11]. A recovery procedure for multiple levels [12], such as global, segment, and local protection, was studied. A network-protection scheme for guaranteeing recovery time [13] was also proposed. In addition, a protection mechanism using fewer network resources by sharing wavelength-division-multiplexing (WDM) resources [14] was issued. By means of this protection mechanism, in the case of multiple network failures, a large number of standby paths are prepared, so a huge volume of physical resources might be needed. It is therefore only useful for limited network failures, such as failures of a few links or nodes.

On the other hand, by means of restoration, recovery paths are calculated one by one after network failures are detected. Restoration schemes for handling multiple failures [15], considering global and local wavelength availability [16], and for virtual networks [17] have also been proposed. In addition, a fast connection-recovery scheme that reduces the search range by using special nodes as a landmark [18] was proposed. This scheme is useful for catastrophic network failures, since all reroutes are basically calculated after the failures are detected. However, if there are a large number of current paths, it might take much time to calculate all recovery paths to the current paths.

III. REQUIREMENTS CONCERNING A NETWORK-DISASTER RECOVERY SYSTEM

The target network structure is shown conceptually in Figure 1. It is composed of an IP network layer and a packet-transport-node (PTN) network layer, such as an MPLS-TP network, controlled by a network-management server (assumed to be connected to all PTNs). The core network is composed of PTNs, while the access network is composed of IP network nodes. In this study, recovery from multiple network failures on the IP and PTN networks (for example, the two network failures shown in the figure) is focused on in this study. One of the critical issues concerning network recovery in the case of a network disaster is the time taken to recover numerous established paths of a packet network. At that time, each path is configured by a label-switched path (LSP) [19] and a pseudo-wire (PW) [20]. Specifically, the main issue is the time taken to recalculate numerous recovery paths one by one after disconnected paths are detected by monitoring network conditions.

In the case of a packet-transport network, the bandwidth of a network path is guaranteed. Therefore, ensuring the quality of a recovery path, such as bandwidth and/or end-to-end delays before (as well as after) a network failure is also an issue.

To tackle the above-mentioned issues, the proposed network-disaster recovery system should satisfy the following four requirements.

- ① Manage multi-layer networks
- ② Recover from multiple network failures
- ③ Rapidly establish recovery paths
- ④ Guarantee quality of recovery paths after network failures are recovered

To meet these requirements, the network-disaster recovery system is designed on the basis of the following policy: If plenty of paths are set up, recovery paths should not be recalculated after multiple network failures are detected (since it takes considerable time to recalculate them). On the other hand, recovery paths that guarantee bandwidths and delays for each possible network failure should be calculated preliminarily, and paths should be promptly recovered by using the prepared paths after the network failures are detected.

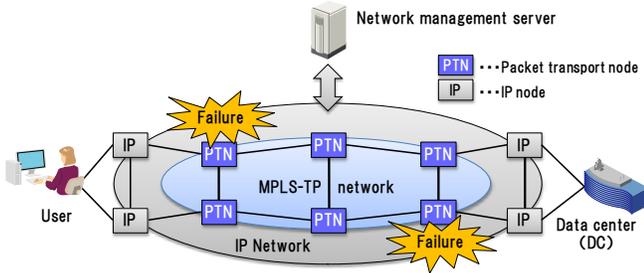


Figure 1. Target network structure

IV. PROPOSED NETWORK-DISASTER RECOVERY SYSTEM

As for the proposed network-disaster recovery system, a network-management server centrally manages an entire network. In the target network, a core-network segment is composed of PTNs, and an access-network segment is composed of IP network nodes. In addition, the network-management server manages the entire network by dividing it into multiple network areas and controlling each of them by using the area-based network-management scheme.

A. Structure of proposed system

The structure of the proposed network-disaster recovery system is shown in Figure 2. As an example of area-based management, the network-management server divides the whole PTN network into eight areas by using a conventional scheme, such as cluster analysis, and manages them by using an area-based management scheme. The eight areas are shown as network (NW) areas (1) to (8) in the figure. In addition, the network-management server is assumed to be connected to all PTNs, a user terminal, and servers in a datacenter (DC) by another management network (not shown in the figure). The network-management server monitors all PTNs, manages available network resources, and keeps them as topology-related data. It also executes swift network-disaster recoveries after detecting catastrophic network failures.

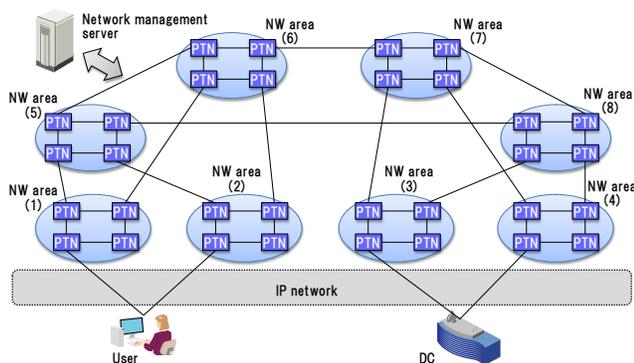


Figure 2. Proposed network-disaster recovery system

As for the proposed disaster-recovery system, the user terminal is connected to a server in the DC via the IP network and PTN networks, and it can get various

application services from the server. To provide the user terminal with robust network access, it is connected to at least two “PTN network areas”. In addition, the DC is connected to at least two other “PTN network areas”.

B. Overview of network-disaster recovery system

The two main procedures used by the proposed network-disaster recovery system are overviewed in Figure 3. Following the first procedure, the network-management server divides an entire PTN network into eight network (NW) areas, labelled (1) to (8) in the figure, and controls them by using the area-based network-management scheme. In addition, it configures the path shown as the solid red lines in the figure as the current path so that the user can access the server in the DC and use application services.

Following the second procedure, the network-management server preliminarily calculates all recovery paths by considering all possible area-based failures. Specifically, the number of possible area-based failure patterns is 255 (since there are eight areas, and each area could be independently active or non-active), namely, 256 (i.e., 2^8) patterns minus a “no network area failure” pattern that is the current network operation. The network-management server assigns a recovery ID for each area-based network failure pattern and stores each recovery ID with information about the recovery paths. It then preliminarily distributes all recovery IDs and the recovery-path information to all PTNs. As stated in Figure 3, it is assumed that network areas (1), (3), and (6) fail. In the case of these failures, the path depicted by a dashed line is prepared as a recovery path, and the recovery-path information is distributed to PTNs related to that recovery path before network operations are started.

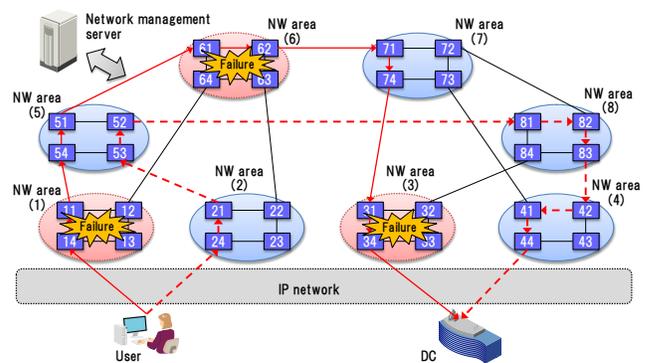


Figure 3. Proposed procedures for network-disaster recovery

During network operations, the network-management server monitors area-based network failures. When it detects an area-based network failure, it determines a failure pattern and a recovery ID. It then distributes the recovery ID to related PTNs, the user terminal, and the server in the DC. The PTNs that receive the recovery ID start to promptly recover and transmit packet data according to the recovery-path information specified by the ID. In addition, the

network-management server configures IP networks to transmit packet data from the user terminal to network area (2). Alternately, if necessary, it transmits a request that asks the user terminal to change the output port so as to transmit packet data to another active network area. Besides, the network-management server configures IP networks to transmit packet data from network area (4) to the server in the DC.

C. Sequence of network-disaster recovery

The proposed network-disaster recovery follows the sequence shown in Figure 4. First, the network-management server divides the entire PTN network into multiple network areas and manages each area by using the area-based network-management scheme, labeled “area mgmt” in Figure 4. Specifically, the PTN networks are divided into eight areas and managed as shown in Figure 3. Subsequently, the network-management server calculates and configures a path as the current path (which is composed of a LSP and a PW) for transmitting packet data from the user terminal to the server in the DC, shown as “current-path configuration” in Figure 4. It starts network operations after configuring the calculated path to related PTNs. As for calculating a path, a route that can provide required bandwidths and transmit packet data within allowed delays is selected as the current path.

The network-management server then calculates and configures all recovery paths, shown as “recovery-path configuration” in Figure 4, by considering all possible area-based network failures. The recovery paths are calculated by a conventional scheme, such as Dijkstra’s algorithm, considering remaining network resources as a backup operation plane for each possible area-based network failure, as shown in Table I. The “recovery-path configuration” shown in the table provides a list of nodes through which data transit. Available network resources are managed by excluding resources belonging to an assumed failure area. Specifically, each recovery path (labeled “P1” in the table) is identified by a recovery ID from “0” to “255.” The top row of the table, containing recovery ID “0”, indicates current-recovery-path configurations for no area-based network failures.

The next row in the table, containing recovery ID “1”, indicates recovery-path configurations for recovering a failure of network area (1). In this case, it is assumed that the network failure occurs in area (1). The recovery path “P1” is calculated on the basis of available network resources. In other words, network resources in area (1) are excluded from the available resources, and the recovery path is calculated. The next row in the table, containing recovery ID “2”, indicates the recovery-path configurations for recovering a failure of network area (2). The row containing recovery ID “38” indicates the recovery-path configurations in the case of failures of network areas (1), (3), and (6). As an example recovery path, the dashed line in Figure 3 is that for the current path depicted by the solid line. In Figure 3 and Table

I, only the recovery-path information for path “P1” is shown as an example. However, the proposed system can manage multiple paths.

As the next step of a recovery, the network-management server calculates recovery-path configurations for each node in each area-based network-failure pattern according to the recovery-path information shown in Table I. As examples, the recovery-path configurations for PTN 53 and 54 are listed in Table II. The information in Table II is obtained by restructuring the node lists of the recovery-path configuration shown in Table I. For example, if PTN 53 is focused in, nodes that are connected to PTN 53 are gathered from Table I and sorted as shown in Table II.

The top row of the table, containing recovery ID “0” on PTN 53, shows the current configuration (i.e., “connection 1” and “connection 2”). With regard to PTN 53, the path P1 (composed of an LSP and a PW) is not configured, since it does not transmit the related packet data. The next row of the table, containing recovery ID “1”, indicates the configuration for recovery path P1 in the case of a failure of network area (1). Specifically, it is shown that PTN 53 transmits packet data of P1 from PTN 21 to PTN 54 and from PTN 54 to PTN 21. In addition, the row of the table containing recovery ID “2” indicates the recovery-path configurations in the case of a failure of network area (2). In this case, the recovery-path configurations for P1 are not included, since PTN 53 does not transmit data for P1. On the other hand, the row of the table containing recovery ID “38” indicates the configurations of recovery path P1 in the case of failures of network areas (1), (3), and (6). Specifically, it is shown that PTN 53 transmits packet data of path P1 from PTN 21 to PTN 52 and from PTN 52 to PTN 21.

In the lower half of the table, recovery-path configurations for PTN 54 are indicated. The row of the table containing recovery ID “0” shows the current configuration. As shown in the table, PTN 54 transmits packet data of path P1 from PTN 11 to PTN 51 and from PTN 51 to PTN 11. The row of the table containing recovery ID “2” indicates the recovery-path configurations in the case of a failure of network area (2). PTN 54 transmits packet data of path 1 from PTN 11 to PTN 51 and from PTN 51 to PTN 11. In addition, the row of the table containing recovery ID “38” indicates the recovery-path configuration in the case of failures of network areas (1), (3), and (6). However, recovery path 1 is not configured, since PTN 54 does not transmit path-1-related packet data. After the network-management server calculates all recovery-path configurations shown in Table II, it distributes them to all PTNs. When each PTN receives the configurations, it stores them with each recovery ID.

In the next step of the recovery sequence, the network-management server monitors operations of all PTNs and area-based network failures, shown as “monitoring” in Figure 4. For example, the network-management server detects the failures of network areas (1), (3), and (6) shown in Figure 3. In this case, the network-management server

selects recovery ID 38 to recover the configured path, shown as “recovery decision”. The PTNs receive recovery ID 38 and configure a data-transmission function to transmit packet data according to the recovery-path information specified by recovery ID 38, shown as “recovery ID distribution”.

In the next step, the network-management server configures IP networks to transmit packet data from the user terminal to PTN 24. In addition, it configures IP networks to transmit packet data from PTN 44 to the server in the DC, shown as “recovery configuration”. In summary, executing the above-described recovery procedures makes it possible to recover failures of network areas (1), (3), and (6).

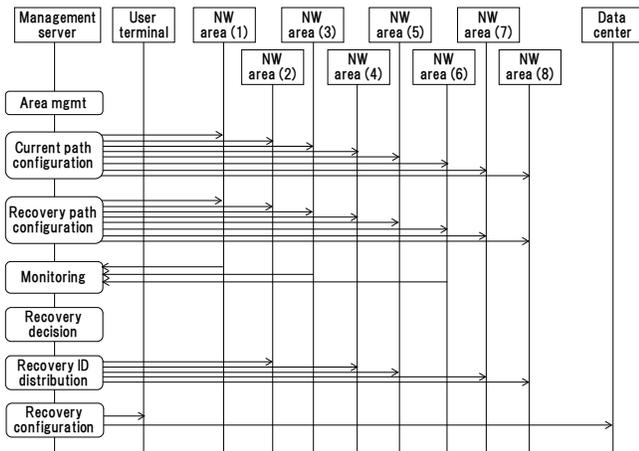


Figure 4. Sequence of network-disaster recovery

TABLE I. RECOVERY-PATH CONFIGURATIONS

Failure pattern	Recovery ID (operation plane)	Path	Recovery path configuration
No failure	0	P1	14, 11, 54, 51, 61, 62, 71, 74, 31, 34
Area (1) failure	1	P1	24, 21, 53, 54, 51, 61, 62, 71, 74, 31, 34
Area (2) failure	2	P1	14, 11, 54, 51, 61, 62, 71, 74, 31, 34
---	---	---	---
Areas (1), (3), (6) failures	38	P1	24, 21, 53, 52, 81, 82, 83, 42, 41, 44
---	---	---	---
All-area failures	255	P1	No recovery

TABLE II. RECOVERY-PATH CONFIGURATIONS FOR EACH PTN

PTN	Recovery ID	Path (LSP/PW)	Connection 1	Connection 2
53	0	---	---	---
	1	P1	21	54
	2	---	---	---
	---	---	---	---
	38	P1	21	52
54	0	P1	11	51
	1	---	---	---
	2	P1	11	51
	---	---	---	---
	38	---	---	---

D. Calculation of recovery paths for possible failure patterns

The flow for calculating a recovery path for an area-based network failure is shown in Figure 5. After the recovery-path calculation starts, delays and available bandwidths between PTNs are calculated from a database that includes topology information and available resources, such as link bandwidths. Next, a possible area-based network failure, for example, a failure of network area (1), is assumed. After that, the PTNs belonging to the assumed network-area failure are excluded from the available resources for calculating recovery paths. After available resources, such as PTNs and bandwidth, are fixed, one of the established PWs is selected to prepare a recovery path. Then, the minimum-delay path that has the same starting and ending points is selected as the recovery path (which is calculated in consideration of available bandwidth and delay). If a recovery path is not found, because of problems like link disconnection, a message indicating “lack of resources” for finding the recovery path is displayed, and the recovery-path calculation process moves on to the next step, namely, selection of another PW. If a recovery path is found, whether it meets the allowed delay time or not is checked. If the path does not meet the allowed delay time, a “lack of available resources” message is displayed, and the calculation process moves on to the next step, namely, finding a recovery path for another PW. If the path meets the allowed delay time, it is determined as the proper recovery path.

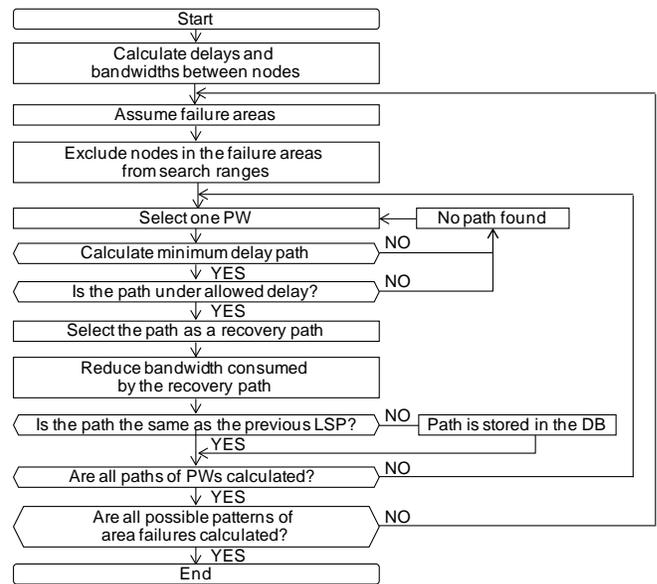


Figure 5. Calculation of recovery paths

After the recovery path is confirmed, available bandwidth is decreased by the amount of bandwidth consumed by the recovery path itself. Subsequently, if the route of the LSP path is not the same as the previously calculated route, it is stored as a new LSP route. Then, whether all recovery paths

for a selected area-based network-failure pattern have been calculated is checked. If all recovery paths are not calculated, the process moves on to the next step, that is, selection of another PW. If all the recovery paths for one area-based network-failure pattern have been calculated, whether all recovery paths for all possible area-based network-failure patterns have been calculated is checked. When all the recovery paths for all possible area-based network failure patterns are calculated, the recovery-path calculation process stops. All recovery paths are calculated, and the recovery-path information is distributed to all network nodes, before network operations are started. Therefore, the nodes can select an appropriate recovery path swiftly when a network fails.

V. IMPLEMENTATION

A prototype system based on the above-described architecture was implemented by using three servers. The configuration of the prototype system—consisting of an application server, a control server, and MPLS-TP simulator server—is shown in Figure 6. Only the structure of the implemented software components is shown in the figure.

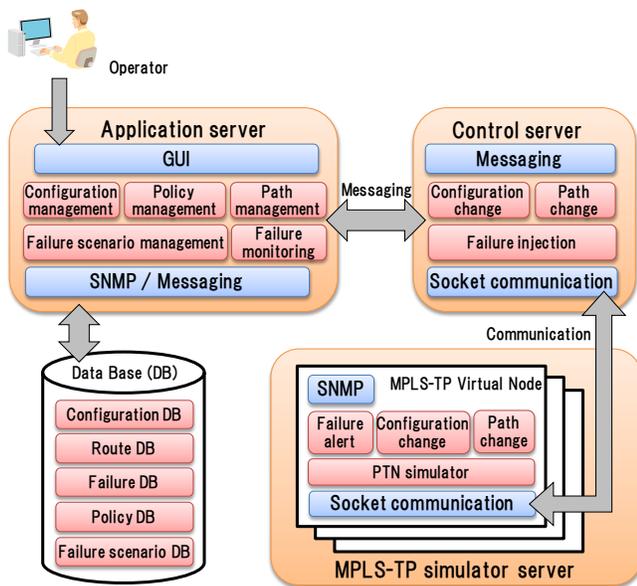


Figure 6. Structure of prototype system

The application server is in charge of overall network management. Specifically, it manages network configurations and policies so that it can select a data-transmission path for each application. In addition, it manages scenarios that would trigger network failures. The control server receives a command message from the application server and transmits it to multiple MPLS-TP virtual nodes. Specifically, it controls the network configuration and data-transmission paths. In addition, it injects network failures according to a failure request from

the application server. The MPLS-TP simulator server simulates certain parts of the MPLS-TP node functions, such as changing the network configuration and setting the LSP and PW paths for data transmission. (Note that it does not simulate real data transmission.) It also detects network failures and transmits alerts to the application server.

A. Recovery procedures executed by the prototype system

The recovery procedures, starting with detecting alerts and finishing with recovering paths, are shown in Figure 7. First, the application server monitors network conditions. When it receives alerts of network-area failures, it analyzes them and updates the network-condition tables. In addition, it updates an alert-history table and indicates the alert on a viewer. It also analyzes the areas in which the failures occurred. The application server then determines whether a network-area failure has occurred. In the prototype system, when all PTNs in an area that receives and transmits data to other areas are damaged, the area is regarded as being in a state of “area failure.” When the application server recognizes several area failures, it determines an area-failure pattern. It then evaluates whether recovery procedures are needed. If no LSP or PW paths are damaged, even if there are area failures, recovery procedures are not taken. If there are damaged paths, recovery procedures are taken. After determining the area-failure pattern, the application server identifies a recovery ID for executing recovery procedures and transmits it to related PTNs. The PTNs change LSP and PW paths according to the path configurations specified by the recovery ID. The records of the LSP and PW paths are updated, and network-area failures are indicated on the viewer.

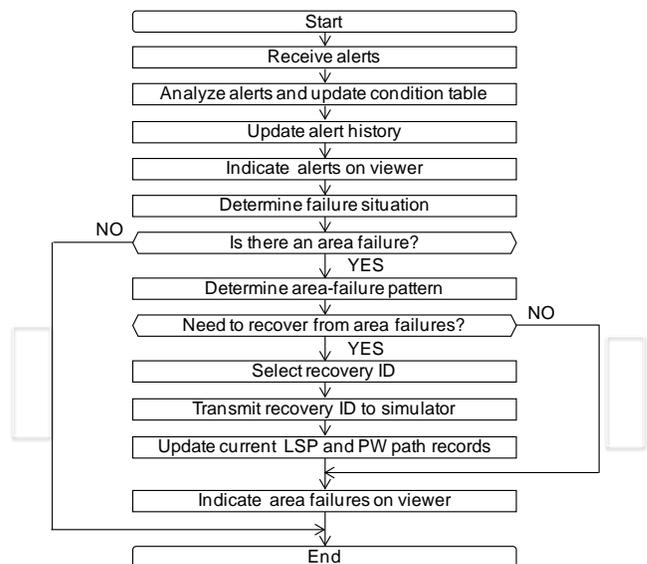


Figure 7. Recovery procedures

B. Implementation of viewer

Viewer functions that enable a user to easily understand operating conditions of the implemented network-disaster recovery system are described in this section.

1) Structure of primary screen

The structure of the primary screen of the prototype system's viewer is shown in Figure 8. The header panel includes a function menu, a user name, a logout button, and so forth. The condition panel displays the current situation regarding networks in certain areas. The topology tree shows a list of connected network nodes in a tree structure. The alert panel indicates up-to-date alerts, showing the level of severity in different colors. The map panel shows the position of the displayed network on the condition panel in relation to the entire network. A network operator can select one of the network-management functions, namely, monitoring network failures, displaying LSP and PW lists, displaying log data of failure histories, setting configurations, and setting failure scenarios, from the pull-down function menu. The "user name" tag shows the name of the current login user. The "logout" button is used to logout of a network-management function.

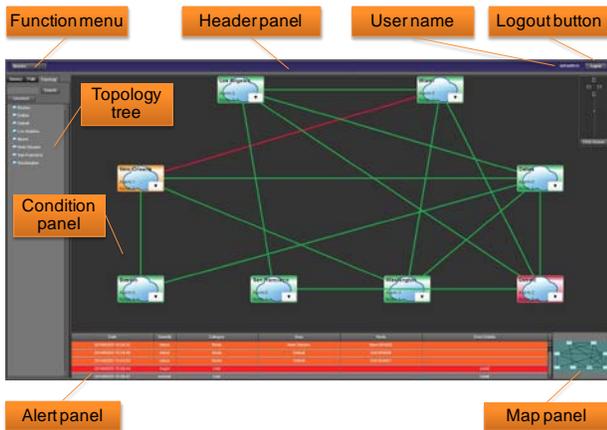


Figure 8. View of primary-screen layout

2) Large-scale-area view

A view of a large-scale area is shown in Figure 9. The "area object" tag shows an area that includes multiple PTNs. In addition, the condition of the PTNs in that area is depicted in different colors. If an area failure has not occurred, the area object is depicted in green. If several PTNs fail, they are depicted in yellow. When an area failure occurs, it is depicted in red. The "area name" tag means the name of the area. The "number of failure nodes" tag means the number of failure PTNs and is written in four digits. On the other hand, the "the number of nodes" tag means the number of all PTNs and is written in four digits. The "area enlarge" button provides a function to show the network topology of the area by a single click. The "link object" tag shows the existence

of a link between areas or between an area and a user. The color of the link is depicted according to one of the following conditions: no failure, partial failure, and full failure. The "zoom palette" button provides various magnifications for viewing the displayed network area. The "scroll" button changes the displayed network area in the direction of a selected button. The "zoom bar" button rescales the displayed network area. The "automatic size control" button resizes the displayed area to its original position.

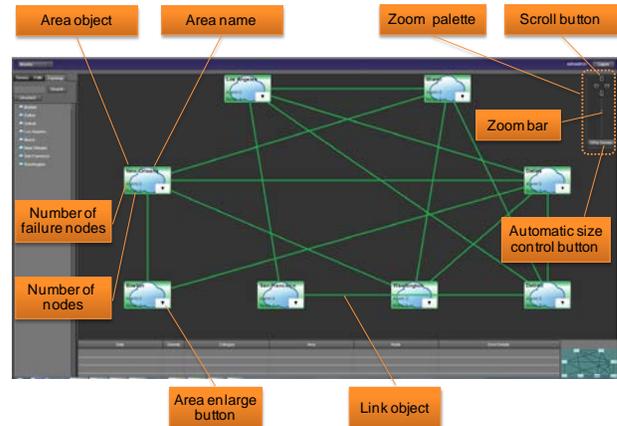


Figure 9. Large-scale area view

3) View of user-connected topology

A view of a user-connected topology is shown in Figure 10. The "list of users" tag shows users connected to the network. The "user object" tag shows an individual user. It includes a "user name" tag showing the name of the current user and a "service name" tag showing the name of the service selected by the user. The "current path highlight" tag indicates the current LSP and PW paths used for communication between users by showing multiple colored-dotted lines. When the "user object" button is clicked, the current path is highlighted for a few seconds.

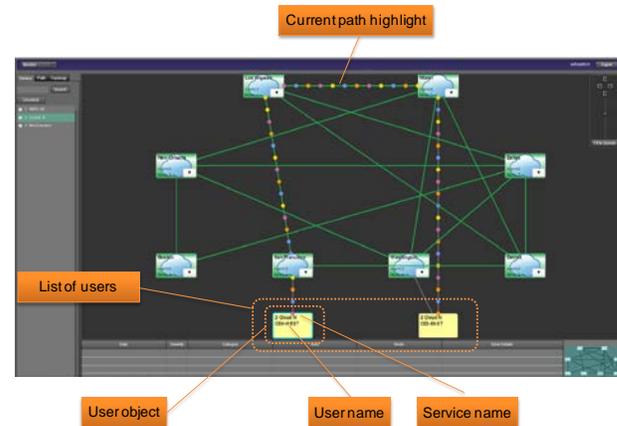


Figure 10. View of user-connected topology

4) Relation between PW and LSP

The relation between a PW and a LSP is shown in the screen view shown in Figure 11. The “user object” tag shows a connection between a user terminal and a PW path. The “PW path” tag shows relations between the user terminal and the LSP path. Each PW path has a unique name. The “edge node” tag indicates a PW edge and is connected to the user and the LSP path edge. On the other hand, the “LSP path” tag shows how the path is structured. Specifically, all PTNs that construct the LSP path are listed. Each PTN object has its own name, and the name of the area that the PTN belongs to is shown in the object. The LSP path also has a unique name, such as its number. The LSP layer is closed by pushing the “close” button.

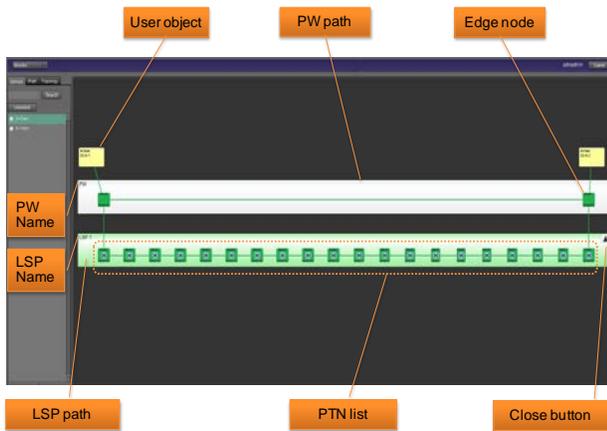


Figure 11. Relation between PW and LSP

VI. PERFORMANCE EVALUATION AND RESULTS

The above-described recovery procedures were evaluated in the case of multiple area-based network failures in networks composed of IP and PTN networks. First, current paths composed of LSPs and PWs were configured to allow users to access application servers in the DCs and use applications provided by the servers. In the evaluation, the procedure for recovering from multiple area-based network failures by using recovery paths was evaluated in terms of whether users can access the application servers. In addition, the time for calculating the current recovery paths and distributing the information concerning the calculated paths to all PTNs was evaluated by changing the numbers of LSPs and PWs used to construct the current paths. Specifically, the case of one user was evaluated in a previous work [1]. In this work, the case of two users was evaluated. In addition, the number of times taken in configuring a recovery path after detecting an area-base network failure was evaluated.

A. Evaluation system

The system used for evaluating the proposed recovery procedures is depicted in Figure 12. It is composed of a network-management server, PTNs, user terminals, and

application servers in DCs. An entire PTN network is divided into eight network areas. Each network area is composed of 12 PTNs, as shown in NW area (7), which is an example network composed of about 100 network nodes. These PTNs are connected in a reticular pattern of 96 PTNs in total. In addition, each user terminal is directly connected to PTN-network areas (1) and (2) by the IP network, and each application server is also connected to PTN-network areas (3) and (4) directly by the IP network.

Note that the PTN networks (composed of 96 PTNs) are simulated by a physical server. In addition, the user terminal and application servers in the DCs are simulated by the same physical server. The specification of the physical server that simulates the PTN networks, user terminals, and application servers is listed in Table III. In addition, another physical server executes the network-management function, but it has the same specifications as the simulator server. In this evaluation, a system composed of eight areas and 96 PTNs is selected since it is large enough to establish a transport core network if ten small packet-transmission nodes, such as an IP node, are connected to each PTN.

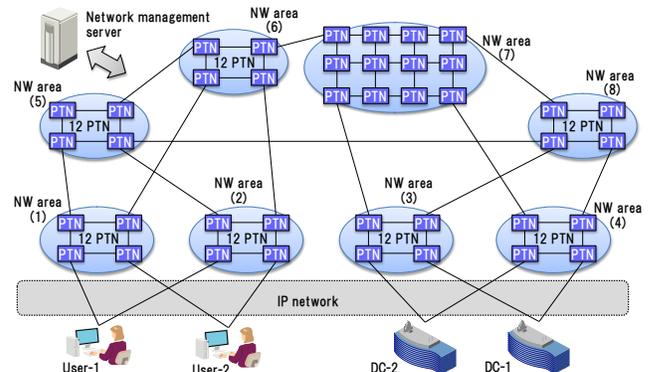


Figure 12. Evaluation system

TABLE III. SPECIFICATIONS OF SERVER

#	Item	Specifications
1	CPU	1.8 GHz, 4 cores
2	Memory	16 GB
3	Storage	600 GB

B. Evaluation condition

The time taken to calculate PWs by using two routes between the users and the application servers in the DCs was evaluated. As an evaluation condition, multiple LSPs between the users and the application servers were established. Each LSP includes 10 PWs (since it usually includes multiple PWs). The evaluations were executed according to the patterns listed in Table IV. Specifically, the time required to calculate current paths and recovery paths

for 255 area-based network-failure patterns was evaluated by changing the number of PWs (namely, 50+50, 250+250, and 500+500) requested by the two users. The time required to distribute all calculated recovery-path configurations and recovery IDs was also evaluated.

TABLE IV. EVALUATION ITEMS

#	Item	Specifications
1	Current-path calculation time	Time taken to calculate 50+50, 250+250, and 500+500 PWs
2	Recovery-path calculation time	Time taken to calculate recovery 50+50, 250+250, and 500+500 PWs for 255 possible area-failure patterns
3	Distribution time	Time taken to distribute all calculated recovery PWs and LSPs for 255 possible area-failure patterns
4	Recovery-ID distribution time	Time taken to distribute a recovery ID after detecting a first area failure

C. Evaluation results

The prototype system was evaluated according to the conditions described by the previous section.

1) Current-path calculation time

The times taken to calculate current PWs requested by the two users are plotted in Figure 13. The evaluation condition is that 10 PWs are included in one LSP. As shown in the figure, the times taken to calculate 100 (50+50) current PWs, 500 (250+250) current PWs, and 1000 (500+500) current PWs were about 64, 326, and 710 milliseconds, respectively.

2) Recovery-path calculation time

The times taken to calculate all recovery PWs for 255 possible area-based network-failure patterns by using one route are plotted in Figure 14. The evaluation condition is that 10 PWs are included in one LSP. As shown in the figure, the time taken to calculate all recovery PWs for 255 area-based network-failure patterns and 100 (50+50) current PWs, 500 (250+250) current PWs, and 1000 (500+500) current PWs are about 5.0, 31.2, and 91.1 seconds, respectively.

3) Distribution time for recovery paths

The times taken to distribute all configurations of calculated recovery PWs to all PTNs are plotted in Figure 15. The evaluation condition is that 10 PWs are included in one LSP. As shown in the figure, the times taken to distribute all configurations of recovery PWs for 255 area-based network-failure patterns and the 100 (50+50) current PWs, 500 (250+250) current PWs, and 1000 (500+500) current PWs are about 239, 315, and 427 milliseconds, respectively.

4) Recovery time from first area-based-network failure

The times taken to distribute the recovery ID to related PTNs and recover from the first area-based network failure for 100 (50+50) current PWs, 500 (250+250) current PWs, and 1000 (500+500) current PWs are plotted in Figure 16. The evaluation condition is that 10 PWs are included in one LSP. Two area-based network-failure patterns, namely, failures of network areas (1), (4), and (6) and failures of network areas (2), (5) and (8), were evaluated since they

include other types of one or two area-based network-failure patterns. As shown in the figure, in the case of failures of network areas (1), (4), and (6), the times taken to recover from the first failure for 100 (50+50) current PWs, 500 (250+250) current PWs, and 1000 (500+500) current PWs are about 596, 1625, and 3332 milliseconds, respectively. On the other hand, in the case of the failures of network areas (2), (5), and (8), the times taken to recover for 100 (50+50) current PWs, 500 (250+250) current PWs, and 1000 (500+500) current PWs are about 649, 1913, and 3513 milliseconds, respectively. As shown in the figure, even if 1000 PWs are setup, the system could recover from the three area failures within four seconds. However, the time taken to recover from the first detected area failure depends on the number of setup PWs. The reason for that dependence seems to be that it takes some time to detect another area failure because the recovery procedures are begun after the first failure is detected. Consequently, the more area-based network failures occur, the longer the time taken to recover from them.

5) Time for recovery-path configuration when number of PWs is changed

The times taken to distribute the recovery ID to related PTNs and setup recovery paths (such as PWs and LSPs) after detecting the first area-based network failure (“configuration time” hereafter) were evaluated (see Figure 17). The evaluation condition is that 10 PWs are included in one LSP. As shown in the figure, the configuration time for 100 recovery PWs, 500 recovery PWs, and 1000 recovery PWs requested by a user are about 70, 94, and 74 milliseconds, respectively. In any case, the configuration time is under 100 milliseconds, even in the case of 1000 PWs. This evaluated time is regarded as the “pure” configuration time for recovery (namely, the time needed for distributing recovery ID and setting up recovery paths, excluding the time taken to detect network failures). In the figure, for a comparison with the proposed method, the time taken to calculate 1000 PWs by a conventional restoration method is also depicted. With the conventional method, calculation of 1000 PWs takes 769 milliseconds. As shown the figure, compared to the proposed method, the conventional method takes much more time to prepare recovery paths.

6) Time for configuring recovery paths when number of LSPs is changed

In the previous experimental evaluation, configuration times for various numbers of setup PWs were evaluated. In this evaluation, the configuration times for various numbers of setup LSPs were evaluated (see Figure 18). Three cases were evaluated. In the first case, namely, 1000 LSPs, a PW is accommodated in each LSP. In the second case, namely, 100 LSPs, 10 PWs are accommodated in a LSP. In the third case, namely, 10 LSPs, 100 PWs are accommodated in a LSP. In all cases, 1000 PWs are setup as recovery paths. As shown in the figure, the configuration times for 1000 current LSPs, 100 current LSPs, and 10 current LSPs requested by a user are about 90, 64, and 64 milliseconds, respectively. In all

cases, the configuration time is under 100 milliseconds, even in the case of 1000 PWs.

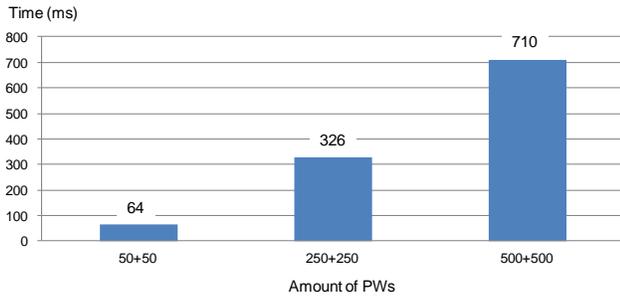


Figure 13. Time for calculating current paths

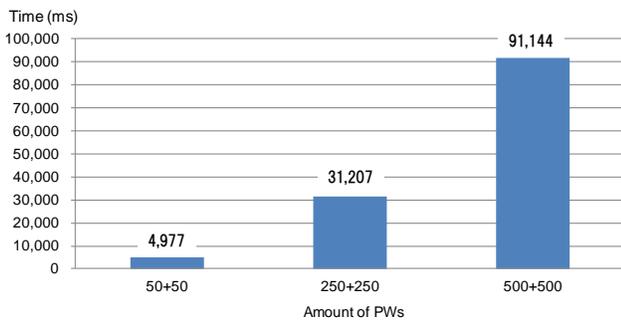


Figure 14. Time for calculating recovery paths

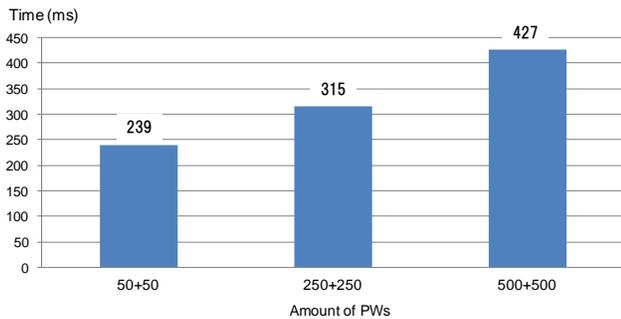


Figure 15. Time for distributing recovery paths

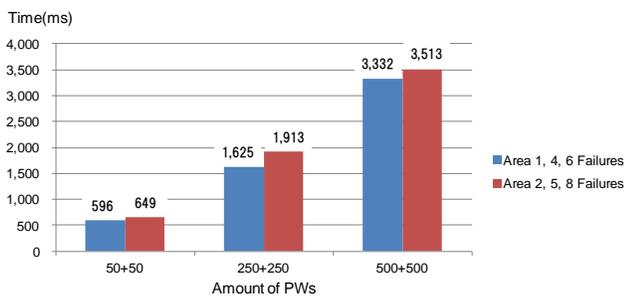


Figure 16. Time for recovery from first area failure

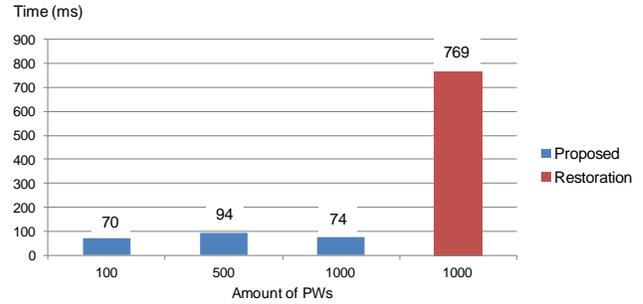


Figure 17. Time for configuring recovery paths when number of PWs is varied

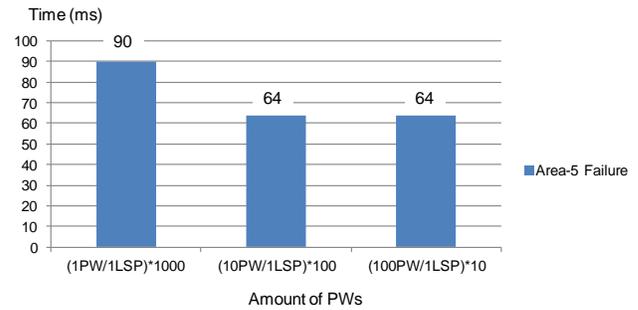


Figure 18. Time for configuring recovery paths when number of LSPs is varied

D. Comparison of proposed system and conventional system

A restoration scheme [2] is basically used when catastrophic network failures occur. In other words, a large number of setup paths are recalculated after network failures are found. According to Figure 13, it takes about 710 milliseconds to calculate paths for 1000 (500+500) PWs. If 1,000,000 PWs exist, it may take 11 minutes and 50 seconds to calculate all the paths. That is, over 10 minutes are needed to calculate recovery paths for the 1,000,000 PWs setup after the network failures were found. On the other hand, in the case of the proposed system, information needed for recovering all the setup paths is distributed to all the network nodes (such as PTNs). According to Figures 17 and 18, the recovery-path configuration time after the network failures are found is less than 100 milliseconds in all cases, since the configuration time is basically independent of the number of setup PWs. Therefore, even if 1,000,000 PWs exist, the proposed system can start recovery within 100 milliseconds after network failures are found.

On the condition that the configuration time depends on the size of the recovery-path database, it takes about 1 minute and 40 seconds to configure 1,000,000 PWs, since 1000 PWs are configured within 100 milliseconds by the proposed system under the conditions specified in Figure 12. Even if the database is very large, recovery-path configurations for all setup paths are selected only once.

Namely, it is enough to select one backup operation plane specified by the recovery ID from the prepared multiple-backup operation planes. It is therefore supposed that the relation between recovery-path configuration time and number of setup paths is almost linear. If 100,000 PWs exist, the proposed system can start to recover paths within 10 seconds. On the other hand, a conventional system based on a restoration scheme takes 1 minute and 12 seconds.

With regard to cost, compared to conventional systems (which use a restoration scheme), the proposed system needs more memory (storage) capacity to keep the recovery paths. As a rough estimation, if the size of the path-configuration data is 1 Kbyte and one-million paths and 1000 backup operation planes exist, each node in the proposed system needs 1 Tbyte of storage (depending on the established paths). However, memory and/or storage costs have been gradually decreasing, so the proposed system is promising for application in the near future.

VII. CONCLUSION

A “network-disaster recovery system” using area-based network management is proposed. As for this system, a whole network is separated into multiple areas. Each area is composed of multiple network nodes, such as MPLS-TP nodes. The system is managed by a network-management server that monitors the condition of every network node and manages the network by detecting area-based failures. Before starting network operations, it calculates recovery paths for every possible area-based failure and distributes them with a recovery ID for each area-failure pattern. The network nodes receive and store the recovery-path configuration and recovery ID. The network-management server detects the network-area failures during network operations and determines a pattern of area failures. Specifically, it determines the numbers and positions of area failures. After determining the pattern of area failures, the network-management server selects an appropriate recovery ID for that pattern and distributes the ID to recovery-related network nodes. The network nodes receive the recovery ID and start data transmission based on the path configuration specified by the distributed ID. After these procedures are completed, the area failures are swiftly recovered.

A prototype system, composed of a network-management server and 96 simulated packet-transport nodes, with a graphical viewer was implemented, and its performance was evaluated. According to the results of the evaluation, all recovery-path configurations for 1000 PWs, namely, transmitting the recovery ID to the related network nodes and using a recovery-path database specified by the ID, are done within 100 milliseconds after network-area failures are detected. On the condition that the configuration time depends on the size of the recovery-path database, the proposed system takes about one minute and 40 seconds in the case of 1,000,000 PWs. On the other hand, it takes a conventional restoration scheme over 10 minutes to calculate

recovery paths under the same evaluation conditions used for the proposed system.

As for the prototype system, the whole network is divided into eight areas as one example of dividing the whole network into multiple area networks. However, the scalability of this approach is an issue. For example, an extended recovery scheme is needed when only one link or node failure occurs, since the proposed system is useful for a large-scale network and multiple failures. The system should be useful for both small failures and large failures. In addition, a consistency of database between a node and a management-server is a future issue. Besides, a recovery procedure is needed in case of a failure of a management server. Therefore, the recovery scheme will be further developed.

ACKNOWLEDGMENTS

Part of this research was included in Research Project O₃ (Open, Organic, Optima) and was supported by the MIC (Japanese Ministry of Internal Affairs and Communications) program, “Research and Development on Virtualized Network Integration Technology”.

REFERENCES

- [1] T. Suzuki et al., “A network-disaster recovery system using area-based network management,” The Third International Conference on Communications, Computation, Networks and Technologies (INNOV 2014), pp. 8-15, Oct. 2014.
- [2] E. Mannie and D. Papadimitriou, “Recovery (protection and restoration) terminology for generalized multi-protocol label switching (GMPLS),” RFC 4427, Mar. 2006.
- [3] International Telecommunication Union - Telecommunication Standardization Sector (ITU-T)
<http://www.itu.int/en/ITU-T/Pages/default.aspx> [retrieved: June, 2015].
- [4] The Internet Engineering Task Force (IETF),
<http://www.ietf.org/> [retrieved: June, 2015].
- [5] B. Niven-Jenkins, D. Brungard, M. Betts, N. Sprecher, and S. Ueno, “Requirements of an MPLS transport profile,” RFC 5654, Sept. 2009.
- [6] M. Bocci, S. Bryant, D. Frost, L. Levrau, and L. Berger, “A framework for MPLS in transport networks,” RFC 5921, July 2010.
- [7] T. Busi and D. Allan, “Operations, administration, and maintenance framework for MPLS-based transport networks,” RFC 6371, Sept. 2011.
- [8] N. Sprecher and A. Farrel, “MPLS transport profile (MPLS-TP) survivability framework,” RFC 6372, Sept. 2011.
- [9] P. Pan, G. Swallow, and A. Atlas, “Fast reroute extensions to RSVP-TE for LSP tunnels,” RFC 4090, May 2005.
- [10] M. Pickavet, P. Demeester, and D. Colle, “Recovery in multilayer optical networks,” *Journal of Lightwave Technology*, Vol. 24, no. 1, pp. 122-134, Jan. 2006.

- [11] J. Zhang, J. Zhou, J. Ren, and B. Wang, "A LDP fast protection switching scheme for concurrent multiple failures in MPLS network," 2009 MINES '09. International Conference on Multimedia Information Networking and Security, pp. 259-262, Nov. 2009.
- [12] Z. Jia and G. Yunfei, "Multiple mode protection switching failure recovery mechanism under MPLS network," 2010 Second International Conference on Modeling, Simulation and Visualization Methods (WMSVM), pp. 289-292, May 2010.
- [13] G. Kuperman and E. Modiano, "Network protection with guaranteed recovery times using recovery domains," INFOCOM, 2013 Proceedings IEEE, pp. 692-700, April 2013.
- [14] J. Rack, "Fast service recovery under shared protection in WDM networks," Journal of Lightwave Technology, Vol. 30, no. 1, pp. 84-95, Jan. 2012.
- [15] M. Lucci, A. Valenti, F. Matera, and D. Del Buono, "Investigation on fast MPLS restoration technique for a GbE wide area transport network: A disaster recovery case," 12th International Conference on Transparent Optical Networks (ICTON), Tu.C3.4, June 2010.
- [16] D. Sheela, M. Smitha Krishnan, and C. Chellamuthu, "Combined link weight based restration strategy in optical networks," 2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 687-690, Mar. 2012.
- [17] T. S. Pham, J. Lattmann, J. Lutton, L. Valeyre, J. Carlier, and D. Nace, "A restoration scheme for virtual networks using switches," 2012 4th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), pp. 800-805, Oct. 2012.
- [18] X. Wang, X. Jiang, C. Nguyen, X. Zhang, and S. Lu, "Fast connection recovery against region failures with landmark-based source routing," 2013 9th International Conference on the Design of Reliable Communication Networks (DRCN), pp. 11-19, Mar. 2013.
- [19] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol lable switching architecture," RFC 3031, Jan. 2001.
- [20] S. Bryant and P. Pate, "Pseudo wire emulation edge-to-edge (PWE3) architecture," RFC 3985, Mar. 2005.



www.iariajournals.org

International Journal On Advances in Intelligent Systems

🔗 issn: 1942-2679

International Journal On Advances in Internet Technology

🔗 issn: 1942-2652

International Journal On Advances in Life Sciences

🔗 issn: 1942-2660

International Journal On Advances in Networks and Services

🔗 issn: 1942-2644

International Journal On Advances in Security

🔗 issn: 1942-2636

International Journal On Advances in Software

🔗 issn: 1942-2628

International Journal On Advances in Systems and Measurements

🔗 issn: 1942-261x

International Journal On Advances in Telecommunications

🔗 issn: 1942-2601