

International Journal on Advances in Internet Technology



The *International Journal on Advances in Internet Technology* is published by IARIA.

ISSN: 1942-2652

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

contact: petre@iaria.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 Internet Technology, issn 1942-2652
vol. 5, no. 1 & 2, year 2012, http://www.iariajournals.org/internet_technology/

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 Internet Technology, issn 1942-2652
vol. 5, no. 1 & 2, year 2012, <start page>:<end page>, http://www.iariajournals.org/internet_technology/

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

Sponsored by IARIA

www.iaria.org

Copyright © 2012 IARIA

Editor-in-Chief

Petre Dini, Concordia University, Canada / China Space Agency Center, China

Editorial Advisory Board

Lasse Berntzen, Vestfold University College - Tonsberg, Norway
Michel Diaz, LAAS, France
Evangelos Kranakis, Carleton University, Canada
Bertrand Mathieu, Orange-ftgroup, France

Editorial Board

Jemal Abawajy, Deakin University, Australia
Chang-Jun Ahn, School of Engineering, Chiba University, Japan
Sultan Aljahdali, Taif University, Saudi Arabia
Shadi Aljawarneh, Isra University, Jordan
Giner Alor Hernández, Instituto Tecnológico de Orizaba, Mexico
Onur Alparslan, Osaka University, Japan
Feda Alshahwan, The University of Surrey, UK
Ioannis Anagnostopoulos, University of Central Greece - Lamia, Greece
M.Ali Aydin, Istanbul University, Turkey
Gilbert Babin, HEC Montréal, Canada
Faouzi Bader, CTTC, Spain
Kambiz Badie, Research Institute for ICT & University of Tehran, Iran
Jasmina Baraković Husić, BH Telecom, Bosnia and Herzegovina
Ataul Bari, University of Western Ontario, Canada
Javier Barria, Imperial College London, UK
Shlomo Berkovsky, NICTA, Australia
Lasse Berntzen, Vestfold University College - Tønsberg, Norway
Nik Bessis, University of Derby, UK
Jun Bi, Tsinghua University, China
Marco Block-Berlitz, Freie Universität Berlin, Germany
Christophe Bobda, University of Arkansas, USA
Alessandro Bogliolo, DiSBef-STI University of Urbino, Italy
Thomas Michael Bohnert, Zurich University of Applied Sciences, Switzerland
Eugen Borcoci, University "Politehnica" of Bucharest, Romania
Luis Borges Gouveia, University Fernando Pessoa, Portugal
Fernando Boronat Seguí, Universidad Politécnica de Valencia, Spain
Mahmoud Boufaïda, Mentouri University - Constantine, Algeria
Christos Bouras, University of Patras, Greece
Agnieszka Brachman, Institute of Informatics, Silesian University of Technology, Gliwice, Poland

Thierry Brouard, Université François Rabelais de Tours, France
Dumitru Dan Burdescu, University of Craiova, Romania
Carlos T. Calafate, Universitat Politècnica de València, Spain
Christian Callegari, University of Pisa, Italy
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Miriam A. M. Capretz, The University of Western Ontario, Canada
Ajay Chakravarthy, University of Southampton IT Innovation Centre, UK
Chin-Chen Chang, Feng Chia University, Taiwan
Ruay-Shiung Chang, National Dong Hwa University, Taiwan
Tzung-Shi Chen, National University of Tainan, Taiwan
Xi Chen, University of Washington, USA
Dickson Chiu, Dickson Computer Systems, Hong Kong
IlKwon Cho, National Information Society Agency, South Korea
Andrzej Chydzinski, Silesian University of Technology, Poland
Noël Crespi, Telecom SudParis, France
Antonio Cuadra-Sanchez, Indra, Spain
Javier Cubo, University of Malaga, Spain
Alfredo Cuzzocrea, University of Calabria, Italy
Jan de Meer, smartspace®lab.eu GmbH, Germany
Sagarmay Deb, Central Queensland University, Australia
Javier Del Ser, Tecnalia Research & Innovation, Spain
Philippe Devienne, LIFL - Université Lille 1 - CNRS, France
Kamil Dimililer, Near East University, Cyprus
Martin Dobler, Vorarlberg University of Applied Sciences, Austria
Eugeni Dodonov, Intel Corporation- Brazil, Brazil
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Matthias Ehmann, Universität Bayreuth, Germany
Tarek El-Bawab, Jackson State University, USA
Nashwa Mamdouh El-Bendary, Arab Academy for Science, Technology, and Maritime Transport, Egypt
Mohamed Dafir El Kettani, ENSIAS - Université Mohammed V-Souissi, Morocco
Armando Ferro, University of the Basque Country (UPV/EHU), Spain
Anders Fongen, Norwegian Defence Research Establishment, Norway
Giancarlo Fortino, University of Calabria, Italy
Kary Främling, Aalto University, Finland
Steffen Fries, Siemens AG, Corporate Technology - Munich, Germany
Ivan Ganchev, University of Limerick, Ireland
Shang Gao, Zhongnan University of Economics and Law, China
Kamini Garg, University of Applied Sciences Southern Switzerland, Lugano, Switzerland
Rosario Giuseppe Garroppo, Dipartimento Ingegneria dell'informazione - Università di Pisa, Italy
Thierry Gayraud, LAAS-CNRS / Université de Toulouse / Université Paul Sabatier, France
Christos K. Georgiadis, University of Macedonia, Greece
Katja Gilly, Universidad Miguel Hernandez, Spain
Feliz Gouveia, Universidade Fernando Pessoa - Porto, Portugal
Kannan Govindan, Crash Avoidance Metrics Partnership (CAMP), USA
Bill Grosky, University of Michigan-Dearborn, USA
Vic Grout, Glyndŵr University, UK

Jason Gu, Singapore University of Technology and Design, Singapore
Christophe Guéret, Vrije Universiteit Amsterdam, Netherlands
Frederic Guidec, IRISA-UBS, Université de Bretagne-Sud, France
Bin Guo, Northwestern Polytechnical University, China
Gerhard Hancke, Royal Holloway / University of London, UK
Arthur Herzog, Technische Universität Darmstadt, Germany
Rattikorn Hewett, Whitacre College of Engineering, Texas Tech University, USA
Nicolas Hidalgo, Yahoo! Research Latin America, France
Quang Hieu Vu, EBTIC, Khalifa University, Arab Emirates
Hiroaki Higaki, Tokyo Denki University, Japan
Eva Hladká, Masaryk University, Czech Republic
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST), Korea
Anna Hristoskova, Ghent University - IBBT, Belgium
Ching-Hsien (Robert) Hsu, Chung Hua University, Taiwan
Christian Hübsch, Institute of Telematics, Karlsruhe Institute of Technology (KIT), Germany
Chi Hung, Tsinghua University, China
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Linda A. Jackson, Michigan State University, USA
Raj Jain, Washington University in St. Louis, USA
Edward Jaser, Princess Sumaya University for Technology - Amman, Jordan
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Georgios Kambourakis, University of the Aegean, Greece
Atsushi Kanai, Hosei University, Japan
Henrik Karstoft, Aarhus University, Denmark
Dimitrios Katsaros, University of Thessaly, Greece
Ayad ali Keshlaf, Newcastle University, UK
Changick Kim, Korea Advanced Institute of Science and Technology, Daejeon, South Korea
Reinhard Klemm, Avaya Labs Research, USA
Samad Kolahi, Unitec Institute Of Technology, New Zealand
Dmitry Korzun, Petrozavodsk State University, Russia / Aalto University, Finland
Evangelos Kranakis, Carleton University - Ottawa, Canada
Slawomir Kuklinski, Warsaw University of Technology, Poland
Andrew Kusiak, The University of Iowa, USA
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Frédéric Le Mouél, University of Lyon, INSA Lyon / INRIA, France
Nicolas Le Sommer, Université Européenne de Bretagne, France
Juong-Sik Lee, Nokia Research Center, USA
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Clement Leung, Hong Kong Baptist University, Hong Kong
Man-Sze Li, IC Focus, UK
Longzhuang Li, Texas A&M University-Corpus Christi, USA
Yaohang Li, Old Dominion University, USA
Jong Chern Lim, University College Dublin, Ireland
Lu Liu, University of Derby, UK
Damon Shing-Min Liu, National Chung Cheng University, Taiwan
Michael D. Logothetis, University of Patras, Greece

Malamati Louta, University of Western Macedonia, Greece
Maode Ma, Nanyang Technological University, Singapore
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Olaf Maennel, Loughborough University, UK
Zoubir Mammeri, IRT - Paul Sabatier University - Toulouse, France
Yong Man, KAIST (Korea advanced Institute of Science and Technology), South Korea
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Chengying Mao, Jiangxi University of Finance and Economics, China
Brandeis H. Marshall, Purdue University, USA
Sergio Martín Gutiérrez, UNED-Spanish University for Distance Education, Spain
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Hamid Mcheick, Université du Québec à Chicoutimi, Canada
Shawn McKee, University of Michigan, USA
Stephanie Meerkamm, Siemens AG in Erlangen, Germany
Kalogiannakis Michail, University of Crete, Greece
Peter Mikulecky, University of Hradec Kralove, Czech Republic
Moeiz Miraoui, Université du Québec/École de Technologie Supérieure - Montréal, Canada
Shahab Mokarizadeh, Royal Institute of Technology (KTH) - Stockholm, Sweden
Mario Montagud Climent, Polytechnic University of Valencia (UPV), Spain
Stefano Montanelli, Università degli Studi di Milano, Italy
Julius Müller, TU- Berlin, Germany
Juan Pedro Muñoz-Gea, Universidad Politécnica de Cartagena, Spain
Krishna Murthy, Global IT Solutions at Quintiles - Raleigh, USA
Alex Ng, University of Ballarat, Australia
Christopher Nguyen, Intel Corp, USA
Vlad Nicolici Georgescu, SP2 Solutions, France
Petros Nicopolitidis, Aristotle University of Thessaloniki, Greece
Carlo Nocentini, Università degli Studi di Firenze, Italy
Federica Paganelli, CNIT - Unit of Research at the University of Florence, Italy
Carlos E. Palau, Universidad Politécnica de Valencia, Spain
Matteo Palmonari, University of Milan-Bicocca, Italy
Ignazio Passero, University of Salerno, Italy
Serena Pastore, INAF - Astronomical Observatory of Padova, Italy
Fredrik Paulsson, Umeå University, Sweden
Rubem Pereira, Liverpool John Moores University, UK
Mark Perry, University of Western Ontario/Faculty of Law/ Faculty of Science - London, Canada
Yulia Ponomarchuk, Far Eastern State Transport University, Russia
Jari Porras, Lappeenranta University of Technology, Finland
Neeli R. Prasad, Aalborg University, Denmark
Drogkaris Prokopios, University of the Aegean, Greece
Emanuel Puschita, Technical University of Cluj-Napoca, Romania
Lucia Rapanotti, The Open University, UK
Gianluca Reali, Università degli Studi di Perugia, Italy
Christoph Reinke, SICK AG, Germany
Jelena Revzina, Transport and Telecommunication Institute, Latvia
Karim Mohammed Rezaul, Glyndwr University, UK

Leon Reznik, Rochester Institute of Technology, USA
Joel Rodrigues, Instituto de Telecomunicações / University of Beira Interior, Portugal
Simon Pietro Romano, University of Napoli Federico II, Italy
Michele Ruta, Politecnico di Bari, Italy
Jorge Sá Silva, University of Coimbra, Portugal
Farzad Salim, Queensland University of Technology, Australia
Sébastien Salva, University of Auvergne, France
Ahmad Tajuddin Samsudin, Telekom Malaysia Research & Development, Malaysia
Josemaria Malgosa Sanahuja, Polytechnic University of Cartagena, Spain
Luis Enrique Sánchez Crespo, Sicaman Nuevas Tecnologías / University of Castilla-La Mancha, Spain
Paul Sant, University of Bedfordshire, UK
Brahmananda Sapkota, University of Twente, The Netherlands
Alberto Schaeffer-Filho, Lancaster University, UK
Peter Schartner, Klagenfurt University, System Security Group, Austria
Rainer Schmidt, Aalen University, Germany
Thomas C. Schmidt, HAW Hamburg, Germany
Didier Sebastien, University of Reunion Island, France
Zary Segall, Chair Professor, Royal Institute of Technology, Sweden
Dimitrios Serpanos, University of Patras and ISI/RC ATHENA, Greece
Jawwad A. Shamsi, FAST-National University of Computer and Emerging Sciences, Karachi, Pakistan
Michael Sheng, The University of Adelaide, Australia
Kazuhiko Shibuya, The Institute of Statistical Mathematics, Japan
Roman Y. Shtykh, Rakuten, Inc., Japan
Patrick Siarry, Université Paris 12 (LiSSi), France
Jose-Luis Sierra-Rodriguez, Complutense University of Madrid, Spain
Simone Silvestri, Sapienza University of Rome, Italy
Åsa Smedberg, Stockholm University, Sweden
Vasco N. G. J. Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Radosveta Sokullu, Ege University, Turkey
José Soler, Technical University of Denmark, Denmark
Boyeon Song, National Institute for Mathematical Sciences, Korea
Victor J. Sosa-Sosa, CINVESTAV-Tamaulipas, Mexico
Dora Souliou, National Technical University of Athens, Greece
João Paulo Sousa, Instituto Politécnico de Bragança, Portugal
Kostas Stamos, Computer Technology Institute & Press "Diophantus" / Technological Educational Institute of Patras, Greece
Vladimir Stantchev, SRH University Berlin, Germany
Tim Strayer, Raytheon BBN Technologies, USA
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Tae-Eung Sung, Korea Institute of Science and Technology Information (KISTI), Korea
Sayed Gholam Hassan Tabatabaei, Isfahan University of Technology, Iran
Yutaka Takahashi, Kyoto University, Japan
Yoshiaki Taniguchi, Osaka University, Japan
Nazif Cihan Tas, Siemens Corporation, Corporate Research and Technology, USA
Terje Jensen, Telenor Group Industrial Development / Norwegian University of Science and Technology, Norway

Alessandro Testa, University of Naples "Federico II" / Institute of High Performance Computing and Networking (ICAR) of National Research Council (CNR), Italy
Stephanie Teufel, University of Fribourg, Switzerland
Parimala Thulasiraman, University of Manitoba, Canada
Pierre Tiako, Langston University, USA
Ioan Toma, STI Innsbruck/University Innsbruck, Austria
Orazio Tomarchio, Università di Catania, Italy
Kurt Tutschku, University of Vienna, Austria
Dominique Vaufreydaz, INRIA and Pierre Mendès-France University, France
Massimo Villari, University of Messina, Italy
Krzysztof Walkowiak, Wrocław University of Technology, Poland
MingXue Wang, Ericsson Ireland Research Lab, Ireland
Wenjing Wang, Blue Coat Systems, Inc., USA
Matthias Wieland, Universität Stuttgart, Institute of Architecture of Application Systems (IAAS), Germany
Bernd E. Wolfinger, University of Hamburg, Germany
Chai Kiat Yeo, Nanyang Technological University, Singapore
Mark Yampolskiy, Vanderbilt University, USA
Abdulrahman Yarali, Murray State University, USA
Mehmet Erkan Yüksel, Istanbul University, Turkey

CONTENTS

pages 1 - 10

An XDI-Based Approach to Represent and Exchange Data Between Federated Clouds

Antonio Celesti, University of Messina, Italy
Francesco Tusa, University of Messina, Italy
Massimo Villari, University of Messina, Italy
Antonio Puliafito, University of Messina, Italy

pages 11 - 25

Virtual Connections in P2P Overlays with DHT-Based Name to Address Resolution

Telesphore Tiendrebeogo, University of Bordeaux, France
Daouda Ahmat, University of Bordeaux, France
Damien Magoni, University of Bordeaux, France
Oumarou Sié, University of Ouagadougou, Burkina Faso

pages 26 - 33

Quality Analysis of a Chaotic Proven Keyed Hash Function

Jacques M. Bahi, FEMTO-ST Institute, UMR 6174 CNRS, Computer Science Laboratory DISC, University of Franche-Comté, France
Jean-François Couchot, FEMTO-ST Institute, UMR 6174 CNRS, Computer Science Laboratory DISC, University of Franche-Comté, France
Christophe Guyeux, FEMTO-ST Institute, UMR 6174 CNRS, Computer Science Laboratory DISC, University of Franche-Comté, France

pages 34 - 43

A Reference Model for Improving an Inter-Organizational IT Management Tool

Mark Yampolskiy, Vanderbilt University, USA
Silvia Knittl, msg systems ag, Germany
Feng Liu, Leibniz Supercomputing Centre (LRZ), Germany

pages 44 - 53

Introducing openBOXware for Android: The Convergence between Mobile Devices and Set-Top Boxes

Lorenz Klopfenstein, STI-DiSBeF - University of Urbino, Italy
Saverio Delpriori, STI-DiSBeF - University of Urbino, Italy
Gioele Luchetti, STI-DiSBeF - University of Urbino, Italy
Andrea Seraghiti, STI-DiSBeF - University of Urbino, Italy
Emanuele Lattanzi, STI-DiSBeF - University of Urbino, Italy
Alessandro Bogliolo, STI-DiSBeF - University of Urbino, Italy

pages 54 - 64

Enhanced QoS and QoE Support in UMTS Cellular Architectures Based on Application Requirements and Core Network Capabilities: An Autonomic Resource Management Perspective

Emanuel Puschita, Technical University of Cluj-Napoca, Romania
Andra Elena Iulia Pastrav, Technical University of Cluj-Napoca, Romania
Cristian Androne, Technical University of Cluj-Napoca, Romania

Alexandru Caruntu, Nokia Romania SRL, Romania

Tudor Palade, Technical University of Cluj-Napoca, Romania

An XDI-Based Approach to Represent and Exchange Data Between Federated Clouds

Antonio Celesti, Francesco Tusa, Massimo Villari and Antonio Puliafito

Dept. of Mathematics, Faculty of Engineering, University of Messina

Contrada di Dio, S. Agata, 98166 Messina, Italy.

e-mail: {acelesti, ftusa, mvillari, apuliafito}@unime.it

Abstract—Cloud providers need to manage and control their assets identifying, retrieving, and exchanging data about their virtual resources in different operating contexts. These tasks are not trivial at all and this leads cloud providers to design proprietary solutions for the management of their virtual resources and services. In this paper, considering IaaS clouds, we discuss an approach based on XDI for the representation of data associated to Virtual Machines (VMs). More specifically, we focus on a scenario including federated clouds renting VMs to other ones, where an exchange of related data is required.

Keywords-Cloud Computing, Federation, Naming System, XDI, Higgings.

I. INTRODUCTION

Nowadays, cloud providers supply many kinds of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) to their users, e.g., common desktop clients, companies, governments, organizations, and other clouds. Such services can be arranged composing and orchestrating several Virtual Environments (VEs) or Virtual Machines (VMs) through hypervisors.

The overwhelming innovation of cloud computing is that cloud platforms can react to events internally rearranging the VMs composing their services pushing down management costs, and the interesting thing is that cloud users are not aware of changes, continuing to use their services without interruptions according to a priori Service Level Agreements (SLAs). For example, when a physical server hosting an hypervisor runs out of resources or is damaged, the cloud can decide to move or “migrate” one or more VMs into another server of the same cloud’s datacenter acting as virtualization infrastructure. Further migrations can be triggered for many other reasons including power saving, service optimization, business strategy, SLA violation, security, etc. In addition, if we consider the perspective of cloud federation where clouds cooperate sharing computational and storage resources, a VM can migrate also into a server of another cloud’s virtualization infrastructure. Another business model which can take place in federated scenarios might be the rent of VEs from a cloud to another.

Such a dynamic and continuously changing scenario involves not only cloud services and VMs, but also other cloud entities such as physical appliances and cloud users. All these entities need to be named and represented both

in human-readable and in machine-readable way. Moreover, they need also to be resolved with appropriate data according to a given execution context. For example, as a VM needs to be identified by a name, it may happen that different entities (e.g., the cloud middleware itself, other federated clouds, cloud administrators, cloud users, etc) may be interested to resolve that name retrieving either data concerning general information on the VM (e.g., CPU, memory, kernel, operating system, virtualization format version, IP address, etc), data regarding processes running inside the VM, or data regarding the performance of the VM (e.g., used CPU and memory usage). In addition, the scenario becomes more complex if we consider the fact that these entities might hold one or more names and identifiers also with different levels of abstraction.

In order to discourage a possible evolving scenario where each cloud based on open source architectures might develop its own proprietary information management system with compatibility problems in the interaction among different cloud name spaces, in our previous work [1], we proposed a standard XRI-based approach for the designing of a seamless cloud naming system able to manage and integrate independent cloud name spaces, extending the OpenXRI libraries [2].

XRI, considered alone, does not support any data interchange mechanism between entities which want to exchange data each other according to their policies. In order to overcome this issue, the OASIS XDI Technical Committee developed the XRI Data Interchange (XDI) [3] technology. In this paper, we discuss how to apply XDI technology, using the Higgings framework [4], for the development of a federated IaaS cloud scenario, where each cloud needs to exchange data with other ones about rented VMs. More specifically, considering several clouds, each one managing its own VMs by means of XRI graphs, we will focus on an use case where a cloud lends VMs to another cloud, thence exchanging the related data (e.g., IP address, how to access the VM, features, performance, etc) in a secure way.

The paper is organized as follows: Section II provides a brief description of cloud name spaces. Section III describes the state of the art of naming systems and the most widely adopted solutions in distributed systems and in ubiquitous computing environments. In Section IV, we provide an

overview of the XDI technology motivating how it suits the management of cloud name spaces and data interchange between federated clouds. In Section V, we describe how to design an XDI-based data management system for a cloud federation scenario using the Higgings framework. In the end, in Section VI, we focus on how to represent resources and users in a cloud using XRI RDF graphs. Conclusions and remarks are summarized in Section VII.

II. CLOUD NAME SPACE ISSUES

In this Section, we briefly summarize the main cloud name space issues which have already been analyzed in [5]. Despite the internal cloud structure, we think cloud entities have many logical representations in various contexts. In addition, there are many abstract, structured entities (e.g., a distributed cloud-service built using other services, each one deployed in a different VE). These entities are characterized by a high-level of dynamism: allocations, changes and deallocations of VEs may occur frequently. Moreover, these entities may have one or more logical representations in one or more contexts. But which are the entities involved in cloud computing? In order to describe such entities, we introduce the generalized concept of *Cloud Named Entity* (CNE). A CNE is a generic entity indicated by a name or an identifier which may refer both to real/abstract and simple/structured entity. As depicted in Figure 1, examples of CNE may be a cloud itself, a cloud federation, a virtualization infrastructure, a server running an hypervisor, a VE, a cloud service, or cloud users including companies, governments, universities, cloud technicians, and desktop clients.

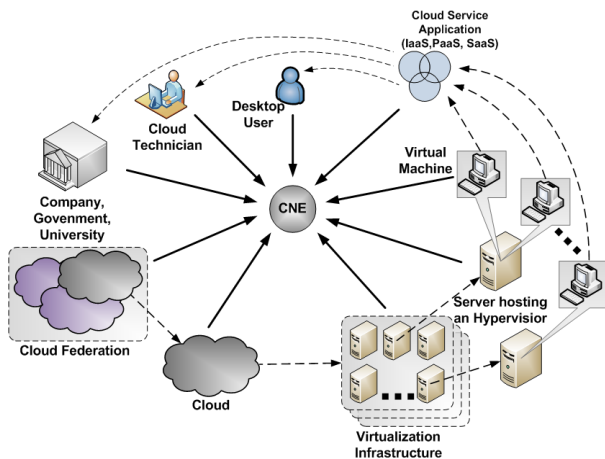


Figure 1. Examples of generic CNEs.

In our abstraction, we assume that a CNE is associated to one or more identifiers. As a CNE is subject to frequent changes holding different representations in various *Cloud Contexts* (CCNTXs), the user-centric identity model [6] seems to be the most convenient approach. We define a

CCNTX as an execution environment where a CNE is represented by one or more identifiers and has to be processed. In this work, we assume a CNE is represented by one or more *CCNTX Resolver Server(s)*, which are servers returning data or services associated to a CNE in a given CCNTX. Figure 2 depicts an example of CNE associated with six identities within four CCNTXs. The target CNE holds identity 1, 2

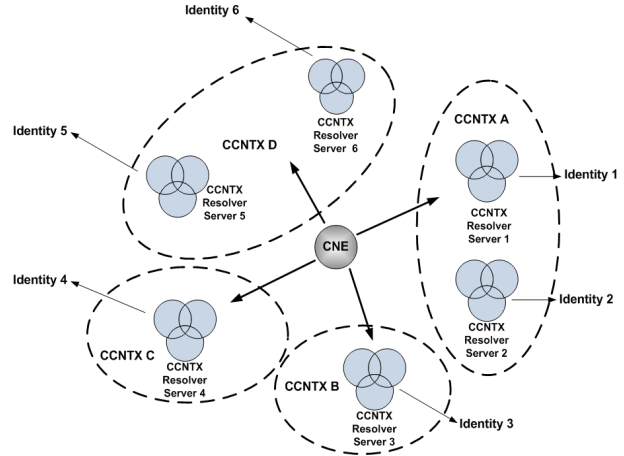


Figure 2. Examples of a generic CNE associated to several CCNTXs.

inside CCNTX A, identity 3 inside CCNTX B, identity 4 inside CCNTX C, and identity 5, 6 inside CCNTX D. We define a *Cloud Naming System* (CNS) as a system that maps one or more identifiers to a CNE. A CNS consists of a set of CNEs, an independent cloud name space, and a mapping between them. A cloud name space is a definition of cloud domain names. Instead, a name or identifier is a label used to identify a CNE. A client resolver which needs to identify a CNE in a given CCNTX performs a resolution task. Resolution is the function of referencing an identifier to a set of data or services describing the CNE in several CCNTXs.

III. RELATED WORK AND BACKGROUND

Cloud computing is generally considered as one of the more challenging research field in the ICT world. It mixes aspects of Utility Computing, Grid Computing, Internet Computing, Autonomic computing and Green computing [7], [8]. Many authors are trying to describe what it exactly means, in terms of Utility Computing (as the Electricity Model, see [9]), its Economics and Benefits, and what are its Obstacles and Opportunities as the TOP 10 list reports in [10], [11]. Cloud, combined with statistical multiplexing, should increase resources utilization compared to traditional data centers, offering services below the costs of medium-sized datacenters and still making good profits (see [12]). In such new emerging environments, even though naming and resource location raise several issues, there have not been many related works in literature yet regarding naming

systems managing cloud name spaces, and DNS is still erroneously considered the “panacea for all ills”. In fact, DNS presents some problems: it is host centric, unsuitable for complex data and services location, and it is not suited to heterogeneous environments. Possible improvements might come from the naming system works in high-dynamic, heterogeneous and ubiquitous environments. An alternative to DNS is presented in [13]. The authors propose a Uniform Resource Name System (URNS), a decentralized solution providing a dynamic and fast resource location system for the resolution of miscellaneous services. Nevertheless, the work lacks of an exhaustive resource description mechanism. With regard to naming system in ubiquitous computing, in [14] the authors propose a naming system framework for smart space environments. The framework aims to integrate P2P independent cloud naming systems with the DNS, but appears unfitted to be exported in other environments. In addition it aims to localize and identify an entity that moves from a smart space to another using as description mechanism the little exhaustive DNS resource records. A hybrid naming system that combines DNS and Distributed Hash Table (DHT) is presented in [15]. The authors adopt a set of gateways executing a dynamic DNS name delegation between DNS resolver and DHT node.

An interesting survey among different technologies for the Resource Discovery in Grid Environments has been done in [16]. The authors presented a valuable comparison among the P2P protocols ranging from Napster, Gnutella, CAN to Chord. It is interesting to notice the punctual evaluation (even taking into account the complexity of each one) of these protocol and their applicability in Grids. They mentioned that one of the main constraints in Grid is the scalability. Some of the protocols reported above are not really fully decentralized. (i.e., Napster) whereas others do not guarantee the operating in heterogeneous Grid environments. Other evaluations were conducted in [17] and [18]. Their assessments are about the possibility to use in Grid consolidated protocols for the Resource Discovery (RD) tasks. However many solutions adopted in Grid ([19]) along with the *advanced DHT* usage (see [20]), are not suitable in clouds at all. We can affirm that the level of heterogeneity in Clouds is higher of any Grid infrastructure. For that reason we cannot consider solutions embraced in Grid, but we have to look solutions widely used in distributed system as Internet (i.e., DNS approach). In our point of view concepts of systems heterogeneity and federation mechanisms need to be taken into account. Whether we consider the recent convergence of Web SSO systems in the Internet, in the last years we assisted to a wide use of OpenID [21]. It is considered as one of the widely digital identity protocol used for making *Federation* among web services. Providers that adopted such a technique range from AOL, BBC, Google, IBM, MySpace, Orange, PayPal, VeriSign, LiveJournal, to Yahoo [22]. The new version of OpenID, 2.0 was released to

overcome some big issues [23]. The way for improving it, is to implement several clauses existing in the XRI Standard Specification [24], [25].

We can assume the XRI standard as a step over of the DNS protocol. All enterprises may continue in using their internal systems for cataloging resources and services, as LDAP, Active Directory (AD), owned database, etc; all these protocols are based on DNS. Our idea is to have an alternative to DNS, a kind of *advanced* DNS protocol, that is XRI, compliant with URI/URL approach able to overcome DNS limitations, also in terms of its representativeness. We can state that XRI might represent an useful abstraction of what already exists in the Internet. In particular we remind the XRI syntax and resolution infrastructure was designed explicitly for Internet-scale digital identity, and we are adopting it for enriching exchanged information in much more complex cloud scenarios maintaining its basic philosophy indispensable for the *Federated Digital Identity* management.

Regarding naming, name resolution, and service location in federated cloud environments, in our previous work [26], we highlighted the involved issues both debating a cloud name space analysis and proposing a generic theoretical cloud naming framework for the management of cloud name spaces. The cloud federation is a scenario where clouds establish a relationship in order to benefit new business advantages [27], for example renting single VM or whole cloud services to other clouds, for example when a cloud run out its computational and storage capabilities or when a cloud needs a service which is not able to allocate. In [28], considering such a cloud naming framework and several use-cases of the European Reservoir Project [29], we performed an analysis of the problems that such use-cases raise regarding the management of cloud name spaces, also debating how the aforementioned cloud naming framework could be adopted to manage naming and service resolution. As possible representation of the cloud naming framework we chose XRI [24] and the eXtensible Resource Descriptor Sequence (XRDS) [25] technologies. The major open source implementation of XRI is OpenXRI [2], which provides a basic Authority Resolution server along with java libraries for the development of XRI-based applications. Another interesting initiative is the Higgings Personal Data Service [4] framework developed by the Eclipse community. Higgings implements the XRI Data Interchange (XDI) [30], i.e., a generalized, extensible service for sharing, linking, and synchronizing structured data over the Internet using XRI-addressable RDF graphs. As well as XRI, XDI is under development by the OASIS [31] Technical Committee.

IV. XDI AND CLOUD COMPUTING

In this Section, after a brief description the XDI technology, we motivate how it can help the cloud name space management and data interchange between federated clouds.

A. XDI Overview

XDI (XRI Data Interchange) is a generalized, extensible service for sharing, linking, and synchronizing structured data over the Internet and other data networks using XRI-addressable RDF graphs. XDI is under development by the OASIS XDI Technical Committee. The main features of XDI are: the ability to link and nest RDF graphs to provide context; full addressability of all nodes in the graph at any level of context; representation of XDI operations as graph statements so authorization can be built into the graph (i.e., a feature called XDI link contracts); standard serialization formats including JSON and XML; and a simple ontology language for defining shared semantics using XDI dictionary services. The XDI protocol can be bound to multiple transport protocols. The XDI TC is defining bindings to HTTP and HTTPS, however it is also exploring bindings to XMPP and potentially directly to TCP/IP. XDI provides a standardized portable authorization format called XDI link contracts. Link contracts are themselves XDI documents (which may be contained in other XDI documents) that enable control over the authority, security, privacy, and rights of shared data to be expressed in a standard machine-readable format and understood by any XDI endpoint. XDI enable to achieve a secure interchange of data between different software entities by means of secure communication channels. These channels can be secured through different techniques, including the Security Assertion Markup Language (SAML), based on the Identity Provider/Service Provider (IdP/SP) model.

RDF graphs are created using XRI, i.e., a standard syntax for identifying entities, regardless any particular concrete representation. The XRI system is similar to DNS, including a set of hierarchical XRI authorities but more powerful. The protocol is built on URI (Uniform Resource Identifiers) and IRI (Internationalized Resource Identifiers) extending their syntactic elements and providing parsing mechanisms. Particular types of URI are URN and URL. Since an URL is also an URI, the protocol provides a parsing mechanism from XRI to URL. Therefore XRI is also compatible with any URN domain. XRI supports persistent and reassignable identifiers by means of i-numbers (Canonical ID) and i-names (Local ID). It also provides four types of synonyms (LocalID, EquivID, CanonicalID, and CanonicalEquivID) to provide robust support for mapping XRIs, IRIs, or URIs to other XRIs, IRIs, or URIs that identify the same target entity. This is particularly useful for discovering and mapping to persistent identifiers as often required by trust infrastructures. XRI enable organization to logically organize entities building XRI RDF graphs. According to the XRI terminology, each entity in the graph is named authority. The protocol provides two additional options for identifying an authority: Global Context Symbols (GCS) and cross-references. Common GCS are “=” for people, “@” for

organization, and “+” for generic concepts. For example the `xri://@XYZ*marketing` indicates the marketing branch of an organization named XYZ, where the “*” marks a delegation.

B. Why Does XDI suit Cloud Computing?

XDI meets the requirements of cloud name space management, data retrieval and data interchange especially in federated cloud environments. With XDI a cloud can keep different RDF graphs representing IaaS, PaaS, and SaaS. In addition, such a technology can be used for both identify and resolve VMs and whole *aaS by means of data retrieval mechanisms. For example, the cloud service provider may need to retrieve three types of information about a VM: general data (e.g., CPU, memory, kernel, operating system); real time performance data (e.g., amount of used CPU and memory used); real time data regarding an internal running process (e.g., the percentage of processed data). Moreover, considering IaaS federated clouds, each provider needs to exchange part of its data with other clouds. For example, let us consider two clouds: A and B. Cloud A, logically organize its own VMs by means of an XRI graph. As Cloud B has run out of resources, it require three VMs to cloud A. So that, cloud A instantiate the three VMs and update its XRI graph. Then, in order to allow cloud B to access the VMs, cloud A, after an authentication of cloud B, discloses how to access the VMs and related data. Authentication can be easily achieved using SAML Single Sign-On (SSO) mechanisms.

In addition, XDI might be used to logically represent instances of composed services. For example, let us consider a service instance composed of several elementary services each one running in a different VM. Thanks to XDI it is possible to create in the graph of a cloud an entry representing the service instance, linking the entries representing the VMs on which the service instance is made up. Other possible applications of XDI can regard for example the management of physical assets, clients, and so on.

V. HOW TO ACHIEVE XDI IN FEDERATED CLOUD USING HIGGINGS

Starting from the considerations of the previous Section, in the following we will point out a concrete scenario of cloud federation, specifically aimed to the IaaS context. In order to address either the problem of sharing information among clouds and achieving their authentication process, we rely on the XDI features. More specifically, our scenario takes advantage of the employment of the Higgings framework, that represents a Personal Data Service (PDS) including the implementation of XDI features.

In the first part of the Section we will introduce and describe the Higgings Framework, whereas in the second part we will present the reference scenario where our solution aims to address the IaaS cloud federation problem.

A. The Higgins Framework

Higgins is an open source project that aims to provide to individuals more control over their personal identity, profile and social network data.

The project is organized into three main areas:

- 1) **Active Clients.** An active client integrates with a browser and runs on a computer or mobile device.
 - *Higgins 1.X:* the active client supports the OASIS IMI protocol and performs the functions of an Information Card selector.
 - *Higgins 2.0:* the plan is to move beyond selector functionality to add support for managing passwords, Higgins relationship cards, as well other protocols such as OpenID. It also becomes a client for the Personal Data Store (see below) and thereby provides a kind of dashboard for personal information and a place to manage “permissioning” deciding who gets access to what slice of the user’s data.
- 2) **Personal Data Store (PDS)** is a new work area under development for Higgins 2.0. A PDS stores local personal data, controls access to remotely hosted personal data, synchronizes personal data to other devices and computers, accessed directly or via a PDS client. It allows the user to share selected aspects of their information with people and organizations that they trust.
- 3) **Identity Services** - Code for (i) an IMI and SAML compatible Identity Provider and (ii) enabling websites to be IMI and OpenID compatible.

B. Reference Scenario

As we have already introduced, in this Section we consider the IaaS cloud federation scenario. In particular, we suppose to have a wide distributed infrastructure, composed of different clouds, belonging to different administrative domains. Each cloud is able to satisfy service requests (in the case of IaaS we consider VMs as resources) coming from its users. When, for some reason (e.g. a temporary load peak) a given cloud is not able to satisfy users’ requests anymore, instead of rejecting them, it could ask the additional needed resources to external providers. These latter might be other clouds able to join the federation.

Obviously, the achievement of this process may be difficult due to several issues that have to be addressed: first of all, when a cloud has expired its resources and ask them to external providers, these have to correctly identify the entity that have generated the request, and accept it only if it has been originated from a trusted source. This leads to the need of managing the authentication process. The simplest solution consists in the possibility of creating a set of credentials for each cloud, on every other cloud aiming to attend the federation. Even though this solution is the

straightforward one, its applicability is limited to a scenario just formed by a small number of entities. If we consider the hypotheses of a growing number of clouds, the creation of credentials for each one may be a different task to manage.

In order to simplify the authentication in such scenarios, the most common adoptable solution might be based on the SSO. Instead of creating lots of credentials for authenticating each cloud on the others, it could be employed a more flexible solution relying on trusted third-parties. This approach minimizes the number of expected credentials, since a given cloud just need to have an account on one (or more) of these third-party to be authenticated on all the entities (clouds) that are trusted with them.

Once the authentication task has been solved, in order to allow resources sharing among the cloud federation entities, a way to organize clouds information is also needed. A cloud service provider, in fact, may need different kinds of information regarding VMs: associated resources, instantaneous workload and internal application state. In a federated environment, part of this information might be shared among the entities taking part to the infrastructure. As we have already pointed out previously, if each cloud stores information by means of an RDF XRI graph, the process of communicating requests for new resources, their allocation on external providers and finally their exploiting will be more flexible and simpler. Section VI will provide more details and examples on RDF XRI graphs generated in our testbed.

For addressing either authentication and information sharing among clouds aiming to federate themselves, we propose the employment of an XDI based framework whose implementation, in our case, relies on Higgins. In the following we provide an overview of the operations involved in the creation of a binding among two different clouds for sharing resources: we will assume that the involved entities are based on Higgins for implementing their features. Once the authentication process is performed using the SSO, the involved clouds will be able to share the needed information using the XRI representation transmitting them over a secure channel created among them.

Assuming the internal cloud organization as depicted in Figure 3, we can consider a three layered stack where in the top part lies the cloud manager layer (that manages all the high-level operations such as authentication, resource discovery etc.).

As the Figure shows, Cloud Manager includes the Resource Manager, which is able to manage resources allocation on external providers if needed through the Cross-cloud Federation Manager component. (for further details see [27]). When a cloud (we call it Cloud A) has expired its own resources and needs to gain them from the outside, a Discovery process is started from the Resource Manager. The result of this task will be a list of external clouds able to satisfy the request. From the retrieved set, it will be

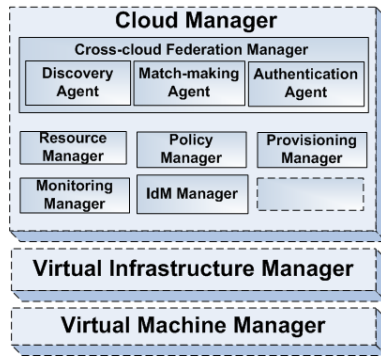


Figure 3. Three layered organization of a Cloud

necessary to select the cloud resource provider best fitting the request: such a task will be accomplished during the Match-Making process. The result of this last operation will consist in the URL of the cloud able to satisfy the resources need of Cloud A (in our case we assume Cloud B as the cloud selected from the Match-Making).

Figure 4 shows the steps involved in the authentication and information sharing between two different clouds (Cloud A and Cloud B): in step 1, the URL coming out from the Match-Making process is used by the Authentication Agent of Cloud A for contacting the destination cloud from which gaining external resources (Cloud B). An HTTP post is forwarded (step 2) with the username and session key (if exists) to the Authentication Agent of Cloud B. The session key is used by the AA of Cloud A as a token proving that a login has been correctly performed on a given IdP and identifies a communication session in a unique way. This means that the key will exist only if the AA already has performed the authentication with an IdP and a session has been created.

During the step 3, the AA of Cloud A is redirected to the IdP trusted with Cloud B for verifying its identity. In step 4, the IdP receives from the AA of Cloud A information about the username and the session key and verifies the existence of a binding between that username and that session key querying a local Database. If an entry exists within the Database, a session for Cloud A has been already created and IdP sends an answer with StatusCode 200, otherwise the login process has to be started for proving the identity of the cloud.

In the last case, in step 5, the AA of Cloud A send its username and password to the IdP that verifies them checking within the LDAP server: if a user exists with that credentials, a new session key is saved within the local Database associated to the username and is sent back to the AA of the Cloud A.

Now that the login phase has been accomplished and the AA of Cloud A holds a valid session key (also registered within the Database), in step 6, it is redirected to the AA of

Cloud B where it is now authenticated and able to perform operations: using its username and its session key, Cloud A can now perform the operations needed for creating VMs instances within Cloud B. In the example reported in Figure 4, the *Add* operation is performed for allocating 3 new VMs.

The AA of Cloud B receives the request and control the associated username and session key: such information are then forwarded to the component that is responsible of managing the XRI graph, which performs the addition of the needed nodes and associate them with the session key associated to the username from which the VMs request is coming. From now on, only the entity that holds that key will be able to access those graph node for retrieving information on the new instantiated VMs. An example of possible XRI graphs of cloud A and B is analyzed in the next Section.

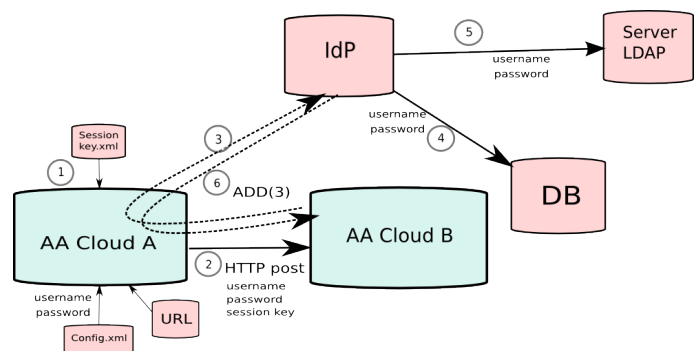


Figure 4. Authentication process and resource information sharing among Cloud A and Cloud B

VI. XRI GRAPH REPRESENTATION OF VMs

In this Section, we show how can be possible to logically organize and manage data associated to a cloud provider by means of XDI and XRI-addressable RDF graphs. More specifically, considering the scenario, we have already pointed out, we discuss how to represent both VMs hosted within the cloud data-center and VMs lent/borrowed to/from other cloud providers. Moreover, for each XRI graphs, we will show the corresponding XDI documents generated in our testbed. XDI allows to represent RDF XRI graph by means of three main elements:

- **Subject**, e.g., `<xdi:s xri="entry"> ... </xdi:s>`. It can be a real/abstract entity represented by means of an XRI entry. Examples can be, the cloud itself, an administrative domain, a cluster, a server, a VM, a cloud-based service instance, an user, etc.
- **Reference**, e.g., `<xdi:ref xri="entry"> ... </xdi:ref>`. It is a reference to a subject.
- **Predicate**, e.g., `<xdi:p xri="relation"> ... </xdi:p>`. It can be relation between two or more subjects.

At the beginning, how depicted in Figure 5, cloud A has an administrative domain including cluster1 and cluster2. Cluster 1 includes server1 which hosts VM1 and VM2,

instead cluster 2 includes server2 which hosts VM3. Cloud A has also two users: user1 and user 2. User1 holds VM1 hosted in server1 of cluster1 and VM3 hosted in server 2 of cluster 2. User 2 holds VM2 hosted in server1 of cluster1.

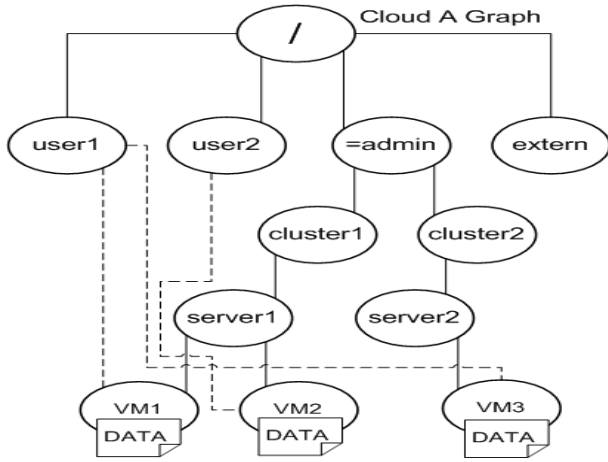


Figure 5. Cloud A graph before federation.

The corresponding XDI code is shown in the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<xdi:xdi xmlns:xdi="http://xdi.oasis-open.org">
  <xdi:s xri="+user1">
    <xdi:p xri="$has$a">
      <xdi:ref xri="+VM1"/>
      <xdi:ref xri="+VM3"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+user2">
    <xdi:p xri="$has$a">
      <xdi:ref xri="+VM2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+extern"/>
  <xdi:s xri="+admin">
    <xdi:p xri="$has">
      <xdi:ref xri="+cluster1"/>
      <xdi:ref xri="+cluster2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster1">
    <xdi:p xri="$has">
      <xdi:ref xri="+server1"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster2">
    <xdi:p xri="$has">
      <xdi:ref xri="+server2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster1+server1">
    <xdi:p xri="$has">
      <xdi:ref xri="+VM1"/>
      <xdi:ref xri="+VM2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster2+server2">
    <xdi:p xri="$has">
      <xdi:ref xri="+VM3"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+VM2">
    <xdi:p xri="DATA">
      <xdi:data><![CDATA[DATA]]></xdi:data>
    </xdi:p>
  </xdi:s>
</xdi:xdi>
```

```
<xdi:s xri="+VM3">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM1">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin+cluster1+server1+VM1"/>
<xdi:s xri="+admin+cluster1+server1+VM2"/>
<xdi:s xri="+admin+cluster2+server2+VM3"/>
</xdi:xdi>
```

At the same time (see Figure 6), cloud B includes cluster1 with server1 and server2. Server1 hosts VM1 and VM2, instead server2 hosts VM3. For simplicity, let us suppose that all VMs are reserved.

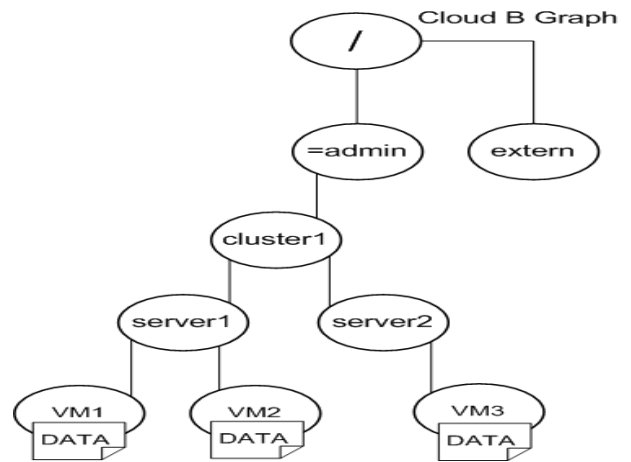


Figure 6. Cloud B graph before federation.

The corresponding XDI code is shown in the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<xdi:xdi xmlns:xdi="http://xdi.oasis-open.org">
  <xdi:s xri="+extern"/>
  <xdi:s xri="+admin">
    <xdi:p xri="$has">
      <xdi:ref xri="+cluster1"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster1">
    <xdi:p xri="$has">
      <xdi:ref xri="+server1"/>
      <xdi:ref xri="+server2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster1+server1">
    <xdi:p xri="$has">
      <xdi:ref xri="+VM1"/>
      <xdi:ref xri="+VM2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+admin+cluster1+server2">
    <xdi:p xri="$has">
      <xdi:ref xri="+VM3"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+VM2">
    <xdi:p xri="DATA">
      <xdi:data><![CDATA[DATA]]></xdi:data>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+VM3">
```

```

<xdi:p xri="DATA">
  <xdi:data><![CDATA[DATA]]></xdi:data>
</xdi:p>
</xdi:s>
<xdi:s xri="+VM1">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="admin+cluster1+server1+VM1"/>
<xdi:s xri="admin+cluster1+server1+VM2"/>
<xdi:s xri="admin+cluster1+server2+VM3"/>
</xdi:xdi>

```

Then, let us suppose that user2 of cloudA requires three additional VMs. As cloud A realizes that it does not have enough resources for instantiate further VMs, it establishes a federation with cloud B, as already described in the previous Section. After authentication, cloudA sends a request for the instantiation of three VMs. So that, cloud B instantiates three VMs in its own datacenter, i.e., VM4 in server1 and VM5 and VM6 in server2. The corresponding updated graph is depicted in Figure 7. Moreover, an user entry for cloud A is created linking the three new instantiated VMs with a reference to them.

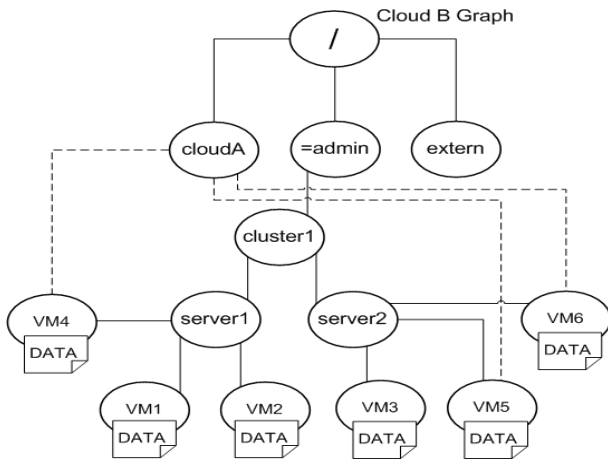


Figure 7. Cloud B graph after federation.

The corresponding XDI code is shown in the following.

```

<?xml version="1.0" encoding="UTF-8"?>
<xdi:xdi xmlns:xdi="http://xdi.oasis-open.org">
  <xdi:s xri="+extern"/>
  <xdi:s xri="admin">
    <xdi:p xri="$has">
      <xdi:ref xri="+cluster1"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="admin+cluster1">
    <xdi:p xri="$has">
      <xdi:ref xri="+server1"/>
      <xdi:ref xri="+server2"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="admin+cluster1+server1">
    <xdi:p xri="$has">
      <xdi:ref xri="+VM1"/>
      <xdi:ref xri="+VM2"/>
      <xdi:ref xri="+VM4"/>
    </xdi:p>
  </xdi:s>
</xdi:xdi>

```

```

<xdi:s xri="admin+cluster1+server2">
  <xdi:p xri="$has">
    <xdi:ref xri="+VM3"/>
    <xdi:ref xri="+VM6"/>
    <xdi:ref xri="+VM5"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM2">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM3">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM1">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="admin+cluster1+server1+VM1"/>
<xdi:s xri="admin+cluster1+server1+VM2"/>
<xdi:s xri="admin+cluster1+server2+VM3"/>
<xdi:s xri="+VM6">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+cloudA">
  <xdi:p xri="$has$a">
    <xdi:ref xri="+VM6"/>
    <xdi:ref xri="+VM5"/>
    <xdi:ref xri="+VM4"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM5">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM4">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
</xdi:xdi>

```

Finally, cloud B sends the data related to the three instantiated VMs (e.g., how to access the VM, IP address, CPU, RAM, storage, operating system, etc) to cloud A which update its XRI graph. For simplicity in the XDI document, we indicated such information with the string "DATA". As depicted in Figure 8, cloud A, under the "extern" entry, creates an entry for cloud B, linking nodes representing the three extern VMs. More specifically, +Ext-VM1, +Ext-VM2, +Ext-VM3 are aliases of the three VMs instantiated in cloud B.

The corresponding XDI code is shown in the following.

```

<?xml version="1.0" encoding="UTF-8"?>
<xdi:xdi xmlns:xdi="http://xdi.oasis-open.org">
  <xdi:s xri="+user1">
    <xdi:p xri="$has$a">
      <xdi:ref xri="+VM1"/>
      <xdi:ref xri="+VM3"/>
    </xdi:p>
  </xdi:s>
  <xdi:s xri="+user2">
    <xdi:p xri="$has$a">
      <xdi:ref xri="+VM2"/>
      <xdi:ref xri="+EXT+VM1"/>
      <xdi:ref xri="+EXT+VM3"/>
      <xdi:ref xri="+EXT+VM2"/>
    </xdi:p>
  </xdi:s>
</xdi:xdi>

```

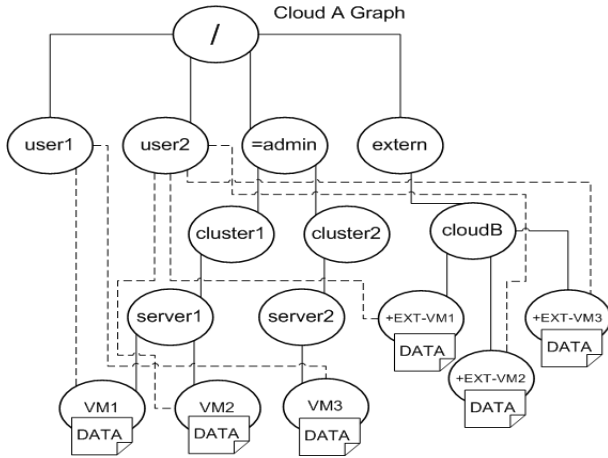



Figure 8. Cloud A graph after federation.

```

</xdi:p>
</xdi:s>
<xdi:s xri="+extern">
  <xdi:p xri="$has">
    <xdi:ref xri="+cloudB"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin">
  <xdi:p xri="$has">
    <xdi:ref xri="+cluster1"/>
    <xdi:ref xri="+cluster2"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin+cluster1">
  <xdi:p xri="$has">
    <xdi:ref xri="+server1"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin+cluster2">
  <xdi:p xri="$has">
    <xdi:ref xri="+server2"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin+cluster1+server1">
  <xdi:p xri="$has">
    <xdi:ref xri="+VM1"/>
    <xdi:ref xri="+VM2"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin+cluster2+server2">
  <xdi:p xri="$has">
    <xdi:ref xri="+VM3"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM2">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM3">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+VM1">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+admin+cluster1+server1+VM1"/>
<xdi:s xri="+admin+cluster1+server1+VM2"/>
<xdi:s xri="+admin+cluster2+server2+VM3"/>
<xdi:s xri="+extern+cloudB">
  <xdi:p xri="$has">

```

```

    <xdi:ref xri="+EXT+VM1"/>
    <xdi:ref xri="+EXT+VM2"/>
    <xdi:ref xri="+EXT+VM3"/>
  </xdi:p>
</xdi:s>
<xdi:s xri="+EXT+VM2">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+EXT+VM3">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+EXT+VM1">
  <xdi:p xri="DATA">
    <xdi:data><![CDATA[DATA]]></xdi:data>
  </xdi:p>
</xdi:s>
<xdi:s xri="+extern+cloudB+EXT+VM1"/>
<xdi:s xri="+extern+cloudB+EXT+VM2"/>
<xdi:s xri="+extern+cloudB+EXT+VM3"/>
</xdi:xdi>

```

In the end, references to entries +Ext-VM1, +Ext-VM2, +Ext-VM3 are linked to user2 who is able to access the three VMs. The interesting thing is that both cloud A and user2 are not aware of the details of where the three VMs are hosted: they can only know the information related to the three VMs. Further information related to cloud B are hidden.

VII. CONCLUSIONS AND REMARKS

In this paper, we focused on how to apply XDI to a federated cloud scenario to represent and exchange data related to cloud-based services. As pointed out, XDI can be used to design several high-level management mechanisms in a cloud system, supporting both data models and security. More specifically, we discuss how can be possible to design XDI-based mechanisms in a cloud system using the Higgings framework focusing on a scenario of federated IaaS clouds lending/borrowing VMs each other. In the end, an use case has been described and implemented, showing the XRI graphs representing VMs and the corresponding XDI documents before and after a federation between two clouds. XDI is a generalized, extensible service for sharing, linking, and synchronizing structured data over the Internet originally thought for web-based systems. In this paper, we hope to success stimulating your interest toward the designing and development of XDI-based mechanisms for cloud computing system. In future works we plan to evaluate the performance of the system also considering the overhead due to the security.

REFERENCES

- [1] M. V. A. P. Antonio Celesti, Francesco Tusa, "Evaluating an open source extensible resource identifier naming system for cloud computing environments," in *The Third International IARIA Conference on Evolving Internet (INTERNET 2011)*, pp. 26–31, June 2011.
- [2] OpenXRI Project, XRI applications and libraries, <http://www.openxri.org/>.

- [3] D. Reed, G. Strongin, XDI (XRI Data Interchange), A White Paper for the OASIS XDI Technical Committee v2, OASIS, 2004.
- [4] "Higgings personal data service, <http://www.eclipse.org/higgings/>."
- [5] A. Celesti, M. Villari, and A. Puliafito, "Ecosystem of cloud naming systems: An approach for the management and integration of independent cloud name spaces," (Los Alamitos, CA, USA), pp. 68–75, IEEE Computer Society, 2010.
- [6] G.-J. Ahn, M. Ko, and M. Shehab, "Privacy-enhanced user-centric identity management," in *IEEE International Conference on Communications, ICC '09*, pp. 14–18, June 2009.
- [7] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, pp. 1–10, 2008.
- [8] R. L. Grossman, "The case for cloud computing," in *IT Professional*, vol. 11, pp. 23–27, March 2009.
- [9] E. Brynjolfsson, P. Hofmann, and J. Jordan, "Cloud computing and electricity: beyond the utility model," *Commun. ACM*, vol. 53, pp. 32–34, May 2010.
- [10] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50–58, April 2010.
- [11] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.
- [12] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599 – 616, 2009.
- [13] D. Yang, Y. Qin, H. Zhang, H. Zhou, and B. Wang, "Urns: A new name service for uniform network resource location," in *Wireless, Mobile and Multimedia Networks, 2006 IET International Conference*, pp. 1–4, 2006.
- [14] Y. Doi, S. Wakayama, M. Ishiyama, S. Ozaki, T. Ishihara, and Y. Uo, "Ecosystem of naming systems: discussions on a framework to induce smart space naming systems development," in *ARES*, p. 7, April 2006.
- [15] Y. Doi, "Dns meets dht: treating massive id resolution using dns over dht," in *Applications and the Internet International Symposium*, pp. 9–15, January 2005.
- [16] S. Chaisiri and P. Uthayopas, "Survey of resource discovery in grid environments," tech. rep., High Performance Computing and Networking Center, Department of Computer Engineering, Faculty of Engineering, Kasetsart University, 50 Phaholyothin Rd., Chatuchak, Bangkok 10900, Thailand, April 2008.
- [17] A. Sharma and S. Bawa, "Comparative analysis of resource discovery approaches in grid computing.," *JCP*, vol. 3, no. 5, pp. 60–64, 2008.
- [18] A. Hameurlain, D. Cokuslu, and K. Erciyes, "Resource discovery in grid systems: a survey," *Int. J. Metadata Semant. Ontologies*, vol. 5, pp. 251–263, July 2010.
- [19] H. Sun, J. Huai, Y. Liu, and R. Buyya, "RCT: A distributed tree for supporting efficient range and multi-attribute queries in grid computing," *Future Gener. Comput. Syst.*, vol. 24, no. 7, pp. 631–643, 2008.
- [20] Y. Mei, X. Dong, W. Wu, S. Guan, and J. Li, "Sdrd: A novel approach to resource discovery in grid environments," in *Advanced Parallel Processing Technologies* (M. Xu, Y. Zhan, J. Cao, and Y. Liu, eds.), vol. 4847 of *Lecture Notes in Computer Science*, pp. 301–312, Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-76837-1_34.
- [21] Wikipedia OpenID, <http://en.wikipedia.org/wiki/OpenID>, July 2011.
- [22] OpenID World Wide Usage, <http://www.ariadne.ac.uk/issue51/powell-recordon/>, June 2007.
- [23] The Security Vulnerability of Reassignable Identifiers, http://dev.inames.net/wiki/XRI_and_OpenID, July 2011.
- [24] Extensible Resource Identifier (XRI) Syntax V2.0, Committee Specification, OASIS, 2005.
- [25] Extensible Resource Identifier (XRI) Resolution V2.0, Committee Draft 03, OASIS, 2008.
- [26] A. Celesti, M. Villari, and A. Puliafito, "Ecosystem of cloud naming systems: An approach for the management and integration of independent cloud name spaces," *IEEE International Symposium on Network Computing and Applications (IEEE NCA10)*, vol. 0, pp. 68–75, 2010.
- [27] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *2010 IEEE 3rd International Conference on Cloud Computing*, pp. 337–345, IEEE, July 2010.
- [28] A. Celesti, M. Villari, and A. Puliafito, "A Naming System Applied to a Reservoir Cloud," in *2010 Sixth International Conference on Information Assurance and Security (IAS)*, pp. 247–252, IEEE, August 2010.
- [29] Resources and Services Virtualization without Barriers (Reservoir) European Project, <http://www.reservoir-fp7.eu/>.
- [30] "Xri data interchange, oasis, <http://wiki.oasis-open.org/xdi/xdigraphmodel>."
- [31] Organization for the Advancement of Structured Information Standards (OASIS), <http://www.oasis-open.org>.

Virtual Connections in P2P Overlays with DHT-Based Name to Address Resolution

Telesphore Tiendrebeogo
LaBRI
University of Bordeaux
Talence, France
Email: tiendreb@labri.fr

Daouda Ahmat
LaBRI
University of Bordeaux
Talence, France
Email: adaouda@labri.fr

Damien Magoni
LaBRI
University of Bordeaux
Talence, France
Email: magoni@labri.fr

Oumarou Sié
LTIC
University of Ouagadougou
Ouagadougou, Burkina Faso
Email: sie@univ-ouaga.bf

Abstract—Current Internet applications are still mainly bound to their transport layer connections. This prevents many features such as end-to-end security and mobility from functioning smoothly in a dynamic network. In this paper, we propose a novel architecture for decoupling communication from their supporting devices. This enforces the complete separation of devices, applications and entities such as users, services and data. Our architecture is based on a peer-to-peer overlay network where each peer has a permanent name and a variable address which depends on its position in the overlay. In order to dynamically map names to addresses, our architecture provides its own distributed hash table system. After presenting the design of our architecture, we provide a scalability analysis and by performing simulations, we assess its efficiency. Simulation results show that our overlay using a name to address resolution based on a distributed hash table is scalable and has acceptable performances given the flexibility it can provide to applications.

Keywords—*overlay; virtual connection; distributed hash table; name resolution;*

I. INTRODUCTION

Current Internet communications are still based on the paradigms set by the TCP/IP protocol stack 30 years ago and they are lacking several key features. Although many efforts have been done during the last decade to provide mobility, security and multicasting, those efforts have mainly been focused on the equipment itself (e.g., computers, smartphones, routers, etc.) rather than on the logical part of the communication. In fact, although we already have a lot of mobile equipment, it is still impossible to transfer a communication from one device to another without interrupting the communication (and thus start it all over again). In the same way, although we have the choice of many applications for carrying one task, it is also still impossible to transfer a communication from one application to another without interrupting the communication. Layer 2 device mobility (e.g., WiFi, WiMAX, 3G and beyond) is nowadays well supported but users still have a very limited access to upper layers mobility (e.g., MobileIP, TCP-Migrate).

In this paper, we propose and describe a new architecture for using virtual connections setup over dynamic peer-to-peer (P2P) overlay networks built on top of the TCP/IP protocol stack of the participating devices. We have named

this architecture CLOAK (Covering Layers Of Abstract Knowledge). This architecture supports names for entities (i.e., users, services, data) and devices, virtual addresses for devices, and virtual sessions for managing all kinds of Internet communications. These new semantics brought by our proposal open up many novel possibilities for such communications. The virtual connections that are setup and managed by our solution, transparently handle the breakdown and restore of transport layer connections (such as TCP or SCTP connections).

This paper is an extended version of our previous work [1]. We have added here a detailed description of the Distributed Hash Table (DHT) mechanism deployed in CLOAK, an analysis of the complexity of the DHT in terms of distances, states and messages, as well as additional simulation results including comparative ones to other existing DHT systems. CLOAK was originally presented in our paper [2] which contained an extensive amount of background and related work as well as some preliminary simulation results upon static networks concerning path length. Improving upon this foundation, our paper [1] presented the protocols and modules of the architecture with greater details and reported simulation results upon dynamic networks concerning routing success ratio, path length and stretch, as well as DHT requests performance indicators. The addressing and routing system based on hyperbolic geometry which is used by CLOAK was presented in our paper [3]. Both the distributed addressing algorithm and the greedy routing algorithm are detailed in this previous paper and we have not included them here to avoid repetition. The implementation of the DHT scheme used by CLOAK over this hyperbolic system is fully explained in Section IV.

The remainder of this paper is organized as follows. Section II presents the design and features of our architecture. Section III describes the main elements of its possible implementation. Section IV presents the binding algorithm used by our DHT. Section V compares the algorithmic complexity of our proposal to those of various existing DHT systems. Section VI presents various results obtained by simulations for evaluating the routing and binding efficiency of our system. Section VII outlines the related previous work done on transport layer mobility, name and address

separation, as well as DHT schemes. We conclude the paper with a summary of our contributions and present our future research directions.

II. ARCHITECTURE

A. Design

In the context of our architecture, a *communication* is a set of interactions between several entities. It can be any form of simplex or duplex communication where information is processed and exchanged between the entities (e.g., talk, view video, check bank account, send mail, etc.). An *interaction* is simply a given type of action carried out between two or more entities by using an application protocol (e.g., FTP, HTTP, etc.). An *entity* is typically a human user but it can also be an automated service such as a server. A communication typically involves a minimum of two entities but it can involve many more in the case of multicast and broadcast communications. Finally, a device is a communication terminal equipment. On the device are running *applications* that are used by an entity to interact with other entities. Given this context, the aim of our architecture is to permit a communication to be carried out without any definitive unwanted interruption when some or all of its components (i.e., device, application or entity) are evolving (i.e., moving or changing) over space and time. Our architecture ensures that a communication has a lifetime that only depends on the will of the currently implied entities. Changes in devices, applications and even entities (when it makes sense) will not terminate the communication.

Figure 1 shows the CLOAK communication paradigm. In order to untie entities, applications and devices, CLOAK introduces the use of a *session*. A session is a communication descriptor that contains everything needed for linking entities, applications and devices together in a flexible way. A session can be viewed as a container storing the identity and the management information of a given communication. Thus the lifetime of a communication between several entities is equal to the lifetime of its corresponding session. As shown on Figure 1, a device can move or be changed for another without terminating the session. Similarly, an application can be changed for another if deemed appropriate or even moved (i.e., mobile code) also without terminating the session. Finally, entities can move or change (i.e., be transferred to another entity) without terminating the session if this is appropriate for a given communication. We can see that in our new architecture, entities, applications and devices are loosely bound together (i.e., represented by yellow arrows in Figure 1) during a communication and all the movements and changes of devices, applications and entities are supported. Note that in Figure 1, only one instance of each part (device, application, entity) of a communication is shown, other instances would obey the same scheme.

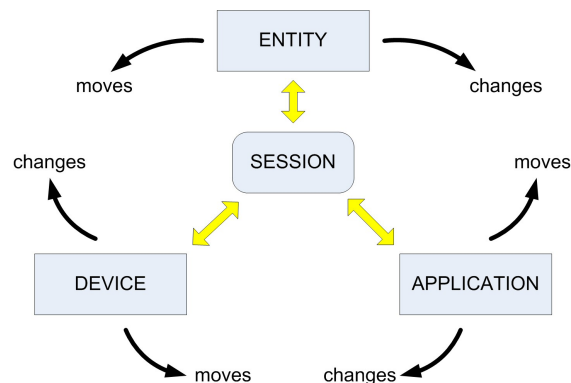


Figure 1. CLOAK communication paradigm.

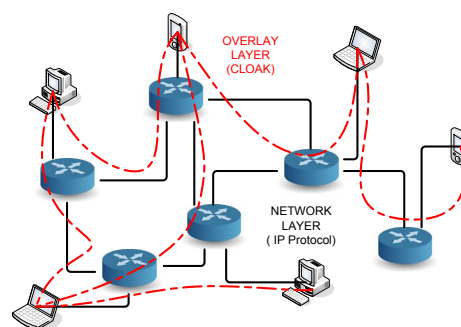


Figure 2. Overlay network.

B. Operation

In order to provide all the above mentioned features, our architecture sets up and maintains a P2P overlay network. Thus, routers are not part of the overlay. Only the devices (i.e., end-hosts or terminals) that wish to share resources in order to benefit from the architecture shall implement and run CLOAK. By doing so, they can join together to form an overlay. Figure 2 shows an overlay example with the links shown in dotted red lines. The devices connect to the others by creating virtual links (upon transport layer connections). Devices with two or more links play the role of overlay routers. The overlay network can build up without any topological constraints, as network devices can connect arbitrarily to each others and join and leave the P2P network at any time.

When joining the overlay, each device obtains a unique overlay address from one of the peers already in the overlay. The method for addressing the peers and routing the packets inside the overlay is based on the groundbreaking work of Kleinberg [4] that assigns addresses equal to coordinates adequately taken from the hyperbolic plane (represented by the Poincaré disk model). His method creates a greedy embedding upon a spanning tree of addresses (named ad-

addressing tree). This addressing tree is a regular tree of degree k . However in Kleinberg's method, the construction of the embedding requires a full knowledge of the graph topology and this topology must also be static. This is required as the degree k of the addressing tree is set to the highest degree found in the network. In our previous work [3], we have enhanced his method in order to manage a dynamic topology which is able to grow and shrink over time. Because we setup an overlay network, we are able to set the degree k of the addressing tree to an arbitrary value and as such, we are able to avoid the discovery of the highest degree node. This specificity renders our method scalable because unlike Kleinberg's method [4], we do not have to make a two-pass algorithm over the whole network to find its highest degree. The fixed degree that we choose determines how many addresses each peer will be able to give. The degree of the addressing tree is therefore set at the creation of the overlay for all its lifetime. In the overlay however, a peer can connect to any other peer at any time in order to obtain an address thus setting the degree does not prevent the overlay to grow. These hyperbolic addresses are appropriately given to the peers so that a greedy routing based on the hyperbolic distance metric is guaranteed to work when the network is connected. Thus, only the addresses of the neighbors of a peer are needed to forward a message to its destination. This is highly scalable as the peers do not need to build and maintain routing tables.

In order to set up the DHT structure needed by our architecture on top of the P2P overlay network, we only need to add a mapping function between a keyspace and the addressing space of the peers. When a peer wants to store an entry in the DHT, it first creates a fixed length key by hashing a key string with the SHA-1 algorithm. Then, the peer maps the key to an angle by a linear transformation. The peer computes a virtual point on the unit circle by using this angle. Next, the peer determines the coordinates of the closest peer to the computed virtual point. The peer then sends a store request to this closest peer. This request is routed inside the overlay by using the greedy routing algorithm presented above.

With the addressing, routing and mapping services provided by our architecture, any user/entity of the P2P overlay network can communicate with any other by setting up a virtual connection on top of the overlay. The steps for establishing a communication between two entities of an overlay are the following:

- 1) Bootstrap into the overlay by setting transport layer connections to one or more devices (i.e., neighbor peers).
- 2) Obtain an overlay address from one of those neighbor peers.
- 3) Identify oneself in the overlay with unique device and entity identifiers.
- 4) Create a session.

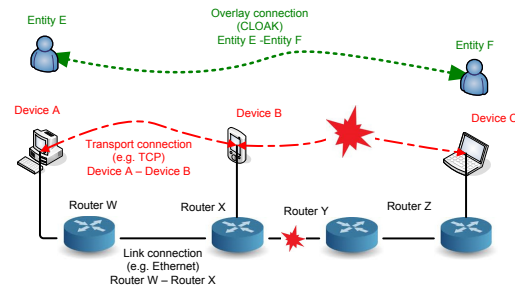


Figure 3. Virtual connections.

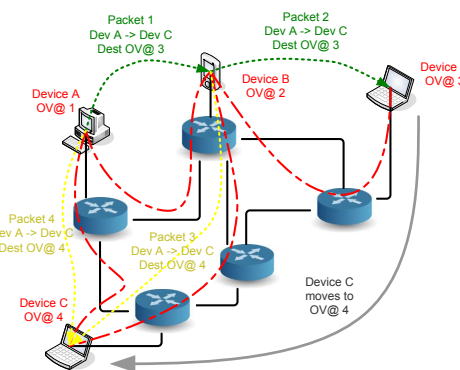


Figure 4. Steering packets inside the overlay network.

- 5) Contact another entity to communicate with inside this session.
- 6) Set an overlay layer virtual connection to this entity as shown in Figure 3.
- 7) Send the data stream through this connection.

If an overlay address becomes invalid, two mechanisms can be used to overcome routing failures. The first one consists, for intermediate nodes, in using the destination name inside the packet header to query the DHT for its new address. If the DHT has a more recent (and thus valid) entry, the intermediate node will then be able to update the header with the new address and forward the packet accordingly. The second one consists, for the destination node, in replacing its old address with its new one in the header of its reply packets. Upon reception, the source node will then be able to update the destination address to the newly received one. These mechanisms are illustrated in Figure 4. We call *steering*, the mechanism of querying the DHT on the fly by intermediate nodes. This mechanism also provides multicast capability when it is performed in each intermediate node. Indeed, a destination group name can be solved as several user names that again can be solved as overlay addresses.

To be able to implement our architecture, we need to introduce several new types of identifiers. More specifically we need to define the following new namespaces:

- Session namespace: any session is attributed a unique identifier that defines the session during its lifetime in the overlay.
- Device namespace: any device is attributed a unique identifier that permanently represents the device. The lifespan of this identifier is equal to the lifespan of its corresponding device.
- Entity namespace: any entity is attributed a unique identifier that represents the entity in a given context. It can be the name of a real person (John Smith) but it could also be the identifier of a professional function (Sales Manager) or the name of an organization (Michelin Company) or a specific service (Areva Accounting service). The lifespan of this identifier is equal to the lifespan of its corresponding entity.
- Application namespace: any application used during a part or all of a session is attributed a unique identifier for receiving data from the other applications of this session. The lifespan of this identifier is equal to the lifespan of the use of the application. If the entity switches to another application, this identifier is updated.

The identifiers will be stored in a DHT built on the P2P overlay network. Each peer will store a fraction of all the records in its naming module. There will be records for the devices (containing pairs like: device ID - overlay address), for the entities (containing pairs like: entity ID - device ID), for the applications (containing pairs like: application ID - session ID) and finally for the sessions (containing pairs like: session ID - session data information). An application using CLOAK will not directly open a connection with an IP address and a port number as with the usual sockets API but it will use the destination's entity ID as well as a stream ID. Figure 5 shows a typical scenario relying on this naming system for solving an entity's location. The yellow oval represents the CLOAK DHT. An entity B registers itself in the DHT by providing the device identifier it is on and its overlay address. Any entity A can now retrieve the location of B by querying the DHT. It can then connect to B via the overlay. When B switches to another device during the same session, A can reconnect to B by using its new overlay address.

As defined earlier, a session is a communication's context container storing everything necessary to bind together entities, applications and devices that are involved in a given communication. Any device, application or entity can be changed or moved without terminating the session. In order to make this possible, the session will be stored in the DHT built by the peers of the overlay network. The DHT will ensure reliability by redundantly storing the sessions on several peers. This session management system ensures the survival of the session until all the entities involved decide to stop it. Figure 6 shows a typical scenario relying on this

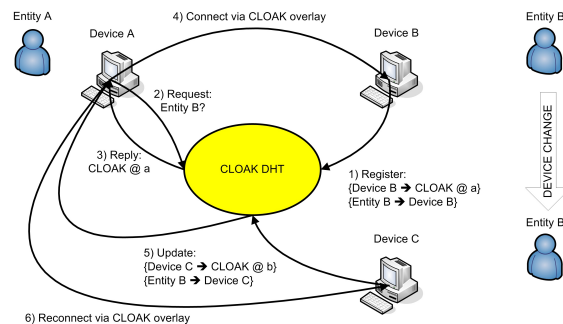


Figure 5. Identification and localization.

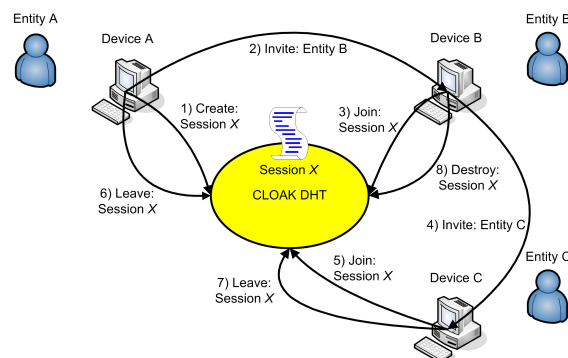


Figure 6. Session management.

session management system. The yellow oval represents the CLOAK DHT. Let us assume that an entity A wants to start a video conference communication with an entity B. It first creates a session called X describing the desired interaction (e.g., video conference) as well as the destination entity that it wants to communicate with (here the entity B). Then A sends an invite message to B that replies by joining the session X. Later on the entity B invites another entity C to participate in the video conference. C accepts and joins the session X. Three entities are now involved in the session X. Later on, the entity A leaves the session X without preventing the others to continue. This thus does not end the session X. Later on the entity C leaves the session X. The entity B being the last one involved decides to destroy the session and thus to end the communication.

C. Usage

Our architecture has a wide range of usages. It provides mechanisms for mobile and switchable applications, for adaptive transport protocol switching and for defining and using new namespaces. It can build scalable and reliable dynamic Virtual Private Networks, define fully isolated Friend-to-Friend networks, or be used as a convergence layer for IPv4, NATs and IPv6. The Table I shows the benefits of *cloaked* applications. Applications are grouped by families. Messaging applications contain e-mail, talk and chat programs. Conferencing applications regroup real-

Table I
FEATURES FOR *cloaked* APPLICATIONS.

Applications	Messaging	Conferencing	Sharing	Streaming
Reachability	✓			
Mobility		✓		✓
E2E privacy	✓	✓		
E2E auth.	✓	✓		
Pseudonymity			✓	
Redirection	✓			✓
Multicasting		✓	✓	✓

time audio and video communications based on signaling protocols such as SIP [5] and H.323 [6]. Sharing applications encompass file-sharing, blogging and social networking applications. Finally, streaming applications contain audio and video broadcasting services such as Internet radios, IPTV, and VoD. Most of the features are usually self-explaining but we give a few examples to highlight possible scenarios. Reachability is the ability to be reached on whatever device the user is currently using. When someone sends a message to an entity, the CLOAK DHT can be used dynamically to determine on which device is the entity and the message is routed to the proper device. Mobility is the ability of CLOAK to hide the handovers of the lower layers to the applications. If an entity is moving or switching devices, real-time applications will be maintained without interruption at the application level. CLOAK can secure connections by using entity IDs rather than device IDs (or IP addresses such as in IPsec), thus establishing End-to-End (E2E) encryption and authentication. The public keys of the peers can be stored in the DHT, however the certification of these keys must be done by a trusted third party. Because CLOAK packets usually transit through several terminals before reaching destination, the IP address of the source is often unknown to the destination thus providing partial pseudonymity. Redirection is the ability to forward a message or a stream to another entity. Finally, multicasting support is provided by CLOAK as group names can be easily set up in the DHT. This feature is useful for saving bandwidth during group communications.

III. IMPLEMENTATION

Figure 7 shows the OSI layers where the CLOAK architecture fits in. CLOAK uses the session layer and the presentation layer between the transport and application layers. These layers do not exist in the Internet stack model but they do already exist in the OSI model. In these two layers we add two new protocols. We add a CLOAK session protocol (CSP) at the session layer and a CLOAK interaction protocol (CIP) at the presentation layer. We also define new identifiers to be used by these new protocols. These new identifiers enable data streams to be bound to entities instead of network identifiers (i.e., IP address, protocol n°, port n°). As shown in Figure 1, identifiers for devices, applications and entities are interwoven together inside a session, but

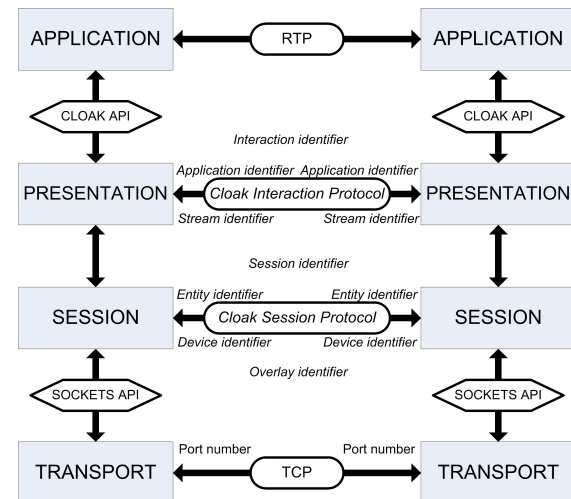


Figure 7. CLOAK architecture in the OSI model.

for the purpose of implementation, we have to order them. We chose to manage a session and its involved devices at the session layer. We also chose to manage the interactions between entities at the presentation layer. As previously said, an interaction is a type of action carried out between two or more entities. It is equal to the use of an existing application layer protocol (e.g., FTP, SMTP, HTTP, etc.). Indeed, our architecture will use the existing application layer protocols as well as the existing transport layer protocols. Thus a file transfer (FTP [7]) client application will still use the FTP protocol to speak to a FTP server. Only the portion of code for establishing a session and thus a connection to the server will have to be rewritten for using the CLOAK API instead of the socket API [8]. The code implementing the application layer protocol will not have to be changed. Please note that the CLOAK API and the mapping of application connections to transport sockets inside the middleware are not defined yet. They will be presented in a future work.

We have shown in Figure 7 how the CLOAK architecture fits in the network protocol stack. We will show how this design translates into the format of the packet headers. Figure 8 shows a CLOAK packet exchanged between a Web client and a Web server. The application header involving the HTTP protocol is now located after the CLOAK headers. We have added two additional headers. The CSP header is located directly above the TCP protocol managing the connection in the operating system of the device. It contains the overlay addresses for routing inside the overlay and enabling device mobility, the device identifiers for switching devices and enabling entity mobility and the entity identifiers for switching entities. The CIP header is located between the CSP and the application level header. It is used for identifying streams and applications. The stream identifiers are used as virtual port numbering on top of the entity. The application identifiers are used for selecting or switching

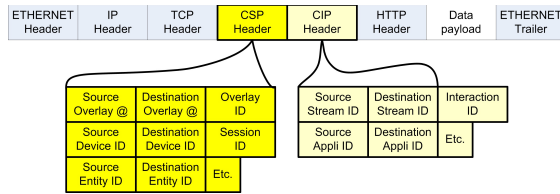


Figure 8. CLOAK protocol encapsulation.

applications when it makes sense in a communication.

The definition and implementation of the CLOAK additional protocols (CSP and CIP) and their corresponding headers will help our architecture to solve some NAT issues because applications using CLOAK will not use IP addresses and ports numbers for setting up or managing connections. They will use unique permanent entity identifiers, thus restoring the end-to-end principle of the Internet communications. The CLOAK architecture will also solve some firewall issues because any type and any number of transport layer connections can be used to connect to a CLOAK overlay. A transport layer connection can act as a multiplex tunnel for the applications using CLOAK. Thus on a given device, the applications can even use only a single port number and a single transport protocol if this is required by the firewall of the device. Indeed, a CLOAK packet has a session ID field and two stream ID fields that enable numerous applications to be multiplexed on a single transport connection if necessary. CLOAK also solves some security issues because security will be implemented by using entity identifiers instead of device identifiers or IP addresses. The security will then be independent from the devices and applications involved. Figure 9 shows the modules composing the CLOAK middleware. The functionalities provided by each module are:

- Bootstrap: primitives for creating a new or joining an existing CLOAK overlay.
- Link: primitives for managing overlay links (i.e., transport layer connections) with the neighbor peers.
- Address: primitives for obtaining an overlay address from an addressing tree parent and for distributing overlay addresses to the addressing tree children.
- Route: primitives for routing the overlay packets with the greedy algorithm using the hyperbolic distance metric.
- Steer: primitives for rerouting overlay packets by using their device or entity identifiers to update their overlay destination address.
- Connect: primitives for establishing and managing overlay virtual connections (i.e., CLOAK layer connections) to other entities.
- Bind: primitives for querying the DHT of the overlay.
- Name: primitives for managing the identifiers used by the peer.

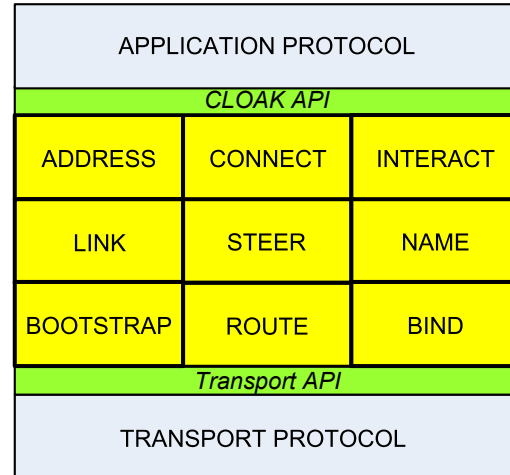


Figure 9. Modules of the middleware.

- Interact: primitives for managing the bindings between the data streams and the applications.

IV. DHT-BASED NAME TO ADDRESS STORAGE

In this section we explain how our overlay system stores and retrieves the (name, address) pairs. Our solution is a structured DHT system that uses the distributed addressing and the greedy routing algorithms presented in our previous work [3].

On startup, each new member of the overlay chooses a name that identifies the device it runs on. This name will be kept by the device during all the lifetime of the overlay. When a new node obtains an address, it stores its name and its address in the DHT, with the name being used as the key and the address as the value. If the same name is already stored in the DHT, an error message is sent back to the node in order to ask the node to select another name. Thus the DHT structure itself ensures that names are unique.

A (key, value) pair is called a *binding*. Figure 10 shows how and where a given binding is stored in the overlay. A binder is any peer that stores these pairs. The depth of a peer in the addressing tree is defined as the number of parent peers to go through for reaching the root of the tree (including the root itself). When the overlay is created, a maximum depth for the potential binders is chosen. This value is defined as the *binding tree depth*. All the peers that have a depth less or equal to the *binding tree depth* in the addressing tree may hold bindings and thus be binders.

When a new peer joins the overlay by connecting to other peers, it obtains an address from one of these peers and it stores its own binding in the system. When a peer wants to store an entry in the DHT, it first creates a key by hashing the name string with the SHA-1 algorithm. It then divides the resulting 160-bit key into r equally sized 160/ r -bit subkeys (for redundancy storage). This r factor is chosen arbitrarily

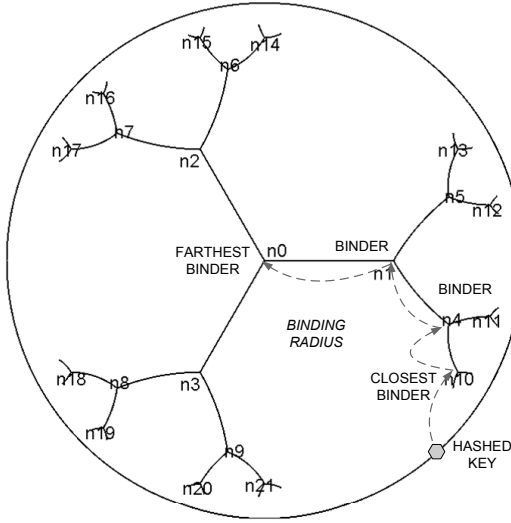


Figure 10. DHT system over the regular spanning tree.

and can be set to whatever value depending on the amount of redundancy required. In absence of redundancy, the peer selects the whole key. Then the (sub)key is mapped to an angle by a linear transformation. The angle is given by:

$$\alpha = 2\pi \times \frac{160/r\text{-bit subkey}}{\underbrace{111\dots 11}_r} \quad (1)$$

The peer then computes a virtual point v on the unit circle by using this angle:

$$v(x, y) \text{ with } \begin{cases} x = \cos(\alpha) \\ y = \sin(\alpha) \end{cases} \quad (2)$$

Next the peer determines the coordinates of the closest binder to the computed virtual point above by using the given *binding tree depth*. In the figure we set the *binding tree depth* to three to avoid cluttering the figure. It's important to note that this closest binder may not exist in reality if no peer is currently owning this address. The peer then sends a store query to this closest peer. This query is routed inside the overlay by using the greedy algorithm presented in our previous work [3]. If the query fails because the binder does not exist or because of node/link failures, it is redirected to the next closest binder which is the father of the computed binder. This process continues until the query reaches an existing binder peer which can be any peer on the path from the computed closest binder to the center peer. Upon reaching an existing binder, the pair is stored in that binder. The query can thus go up the addressing tree to the center peer having the address (0;0) which is the farthest binder. The path from the computed closest binder to the farthest binder is defined as the *binding radius* because it is a shortest path from the edge of the disk to its center.

This process ensures that the pairs are always stored first in the binders closer to the unit circle and last in the binders closer to the disk center. If the addressing tree is imbalanced, many pairs may be stored in peers close to the center thus overloading them. In order to solve this issue any binder peer will be able to set a maximum number of stored pairs and any new pair to store will be rejected and the query redirected as above. Furthermore, to provide redundancy, the peer can repeat the storing process described above for each of the other $r-1$ subkeys. Thus r different binding radiuses can be used and this will improve the evenly distribution of the pairs. In addition, and still for redundancy purposes, a pair may be stored in more than one peer of the binding radius. A binder could store a pair and still redirect its query for storing it in its other ancestor binders. The number of stored copies of a pair along the binding radius may be an arbitrary value set at the overlay creation. We have thus defined two redundancy mechanisms for storing copies of a given binding:

- 1) We can use one or more binding radius(es) by creating r uniformly distributed subkeys.
- 2) We can store the pair in one or more binder(s) of the same binding radius.

These mechanisms enable our DHT system to cope with an non-uniform growth of the overlay and they ensure that a pair will be stored in a redundant way that will maximize the success rate of its retrieval. The number r of subkeys and the number of copies in a given radius are parameters that can be set at the creation of the overlay. Increasing them leads to a tradeoff between improved reliability and storage space cost in binders.

Our solution has the property of consistent hashing: if one peer fails, only its pairs are lost but the other binders are not impacted and the whole system remains coherent. However, this property does not hold true when a partial readdressing takes place as explained in our previous paper [3]. In this case, all the pairs stored in the peers having addresses derived from the failed or unreachable peer's address are lost. To solve this issue, as in many existing systems, pairs will be stored by following a hybrid soft and hard state strategy. Thus a pair will have to be stored by its creator every δt period of time otherwise it will be flushed by the binders that store it. These periodic store messages will ensure that pairs lost by a partial readdressing will be restored after at most δt period of time. A delete-message may be sent by the creator to remove the pair before the end of the period. We analyze the influence of the degree of the addressing tree on the query success rate and the query path length in Section VI.

V. SCALABILITY ANALYSIS OF OUR DHT SYSTEM

We provide in this section a brief complexity analysis of our proposal and compare the results with other existing DHT systems. We first define the four metrics that we use

in our analysis. These metrics were defined and used in the survey of Lua *et al.* [9].

- 1) Hops: this metric counts the average number of peers to go through to reach the destination. It is also named: path length, routing distance or dilation.
- 2) Paths: this metric counts the average number of paths that are crossing any given peer. It is also named congestion.
- 3) States: this metric counts the number of states that must be stored in a peer for the routing to work, it is typically equal to the number of entries found in the routing table of the peer. It is also named routing or memory states.
- 4) Churn messages: this metric counts the average number of messages that are exchanged when a peer joins or leaves the overlay. It is also named join/leave peers or linkage.

In our system, the peers in the overlay connect to each others as they wish, thus no strict topology is enforced. Any peer can have as many links as it can with other peers and one link is of course a minimum to connect to the overlay. The only requirement is that the embedded addressing tree which is a spanning tree of the overlay shall remain valid for the greedy routing to work.

Because any overlay will be at least (when no redundant links exist) composed of its addressing tree, the distances between any two nodes are expected to be of the order of $O(\log(n))$ hops. If the peers have a large number of redundant links (i.e., links not belonging to the addressing tree), the distances will be much shorter. If the overlay topology takes the form of a scale free network [10], the distances will be the order of $O(\log(\log(n)))$ as shown in [11]. Whatever the topology, the number of paths crossing any one peer (its congestion level) will have an expected probability of at most $O(\log(n)/n)$.

When a peer joins the overlay, only its neighbors (i.e., those having setup a link with the new peer) need to update their state information which bears a message cost complexity independent of n . Similarly, when a peer leaves the overlay, only its neighbors need to update their state information also giving a message complexity cost being of the order of $O(1)$. However, if the addressing tree is broken and cannot be restored in a reasonable amount of time as explained in our previous paper [3], a partial readdressing can occur for peers having addresses derived from the failed or unreachable peer's address. In this latter case, which is expected to be very uncommon, the message cost complexity is expected to be of the order of $O(n)$.

Readdressing is needed to provide to the peers the ability of connecting to whatever peers they want. If we force some peers to connect to some specific peers for restoring the addressing tree (as done by Chord, where a peer's IP address determines to which peers it must connect) then the message cost complexity is expected to be of the order of $O(1)$ for

Table II
EXPECTED PERFORMANCE MEASURES OF VARIOUS DHT SYSTEMS.

Lookup	Hops	Paths	States	Churn messages
CAN	$O(n^{(1/d)})$	$O(n^{(1/d)}/n)$	$O(1)$	$O(1)$
Chord	$O(\log(n))$	$O(\log(n)/n)$	$O(\log(n))$	$O(\log^2(n))$
CLOAK	$O(\log(n))$	$O(\log(n)/n)$	$O(1)$	$O(1)/O(n)$
Kademlia	$O(\log(n))$	$O(\log(n)/n)$	$O(\log(n))$	$O(\log(n))$
Pastry	$O(\log(n))$	$O(\log(n)/n)$	$O(\log(n))$	$O(\log(n))$

a leaving peer. Thus readdressing must be seen as a costly feature that can be opted out if performance is desired over flexibility.

Because we use greedy routing, we do not construct and maintain routing tables and the number of states to maintain in any one peer is only equal to the number of its neighbor peers which does not grow with n thus giving a constant complexity cost being of the order of $O(1)$.

Table II compares the complexity costs of the four above defined metrics of various DHT systems including our solution. For CAN, d is an integer equal to or greater than 2 and thus $0 < 1/d < 1$. The results presented in this table have been gathered by using the data published in [9] as well as from our previous analysis. Note that log functions with different constant bases are considered equivalent.

VI. SIMULATIONS

In this section, we present the preliminary results of the simulations that we have carried out to establish a proof-of-concept of our dynamic P2P overlay architecture. We have used our packet-driven discrete event network simulator called *nem* [12] for obtaining all the results shown in this paper.

A. Parameters

In order to evaluate our overlay system on a realistic topology, we have used a 4k-node IP level Internet map created from real data measurements with the *nec* software [13]. In all simulations, the first peer creating the overlay is always a randomly picked node of the map. We have considered that only some nodes of a map at any given time are acting as overlay peers. The simulator's engine manages a simulation time and each overlay peer starts at a given time for a given duration on a random node of the map. The peer that creates the overlay remains active for all the duration of a simulation. The packets are delivered between the nodes by taking the transmission time of the links into account. Peers bootstrap by contacting the node that holds the peer that created the overlay, search for other peers to which they can connect, obtain an address from one of the peers they are connected to and send data or requests messages. This process models the birth, life and death of the overlay.

In any dynamic simulation, there is a warm up phase at the beginning and a cool down phase at the end that must both be considered as transitory regimes. Indeed, at the beginning

only the creator peer exists before new peers start and join it. Similarly, at the end, all peers are gradually leaving the overlay until only the creator peer is left and then it stops. Each simulation runs for 1 hour, thus only measurements in the middle of the simulation (around 30 minutes) can be considered as representing a steady state regime. This comment must be taken into account when looking at all the plots below as most of them show a curve with a typical plateau in the middle. The most significant measurements are those located in this flat part of the plots although the other measurements are also valid.

The number of new peers is set to 30 per minute with random inter-arrival times set with a probability following an exponential distribution. Each peer has a random lifetime set with a probability following an exponential distribution with $\lambda = 10e - 5$ which gives a median value of 300 seconds and a 90th percentile value of 1000 seconds. As each dynamic simulation lasts for 1 hour, this distribution of the peers' session lengths produces a lot of churn. The peers create overlay links with other peers by selecting those which are closer in terms of network hops. Finally, we collect measurements every 600 seconds.

B. Results

We evaluate here the performances of the overlay routing depending on the chosen fixed addressing tree degree as explained in II-B. Data packets are sent by each peer at a rate of 1 every 10 seconds. We only want to evaluate routing success, query success and path lengths but not bandwidth or throughput for now that is why we do not use more realistic generated traffic patterns. The routing success rate for a given peer is equal to the number of data packets properly received by their destinations divided by those sent by the peer. Each point shown on the following graphs is the average value of 20 runs, and the associated standard deviation values are plotted as error bars. We observe the average routing success rate, the average path length and the 90th percentile path length as a function of the addressing tree degree of the overlay. In Figure 11, we can see that the routing success rate is always above 90% which confirms the proper functioning of our system which maintains a high routing success rate despite the churn.

Figure 12 shows the average path length of the hyperbolic routing. The path length is measured as the number of IP hops covered by the packet from the source peer to the destination peer. We can see that values are larger than the ones measured in the static simulations presented in our previous work [3] because here only a subset of the nodes are peers belonging to the overlay thus statistically increasing the distances. In the static simulations, the paths from all pairs were evaluated and the overlay topology was the same as the map itself. Here the nodes form an overlay which may have a different topology and thus lower path length optimality. This remains true even though overlay

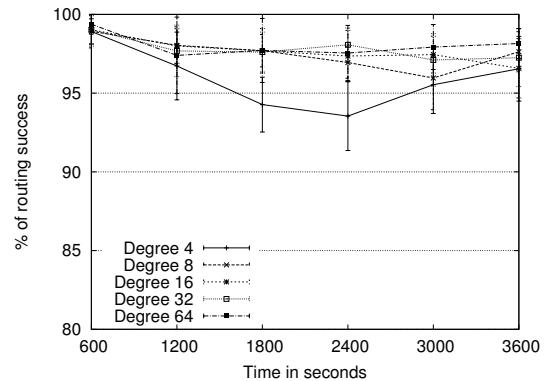


Figure 11. Average routing success rate.

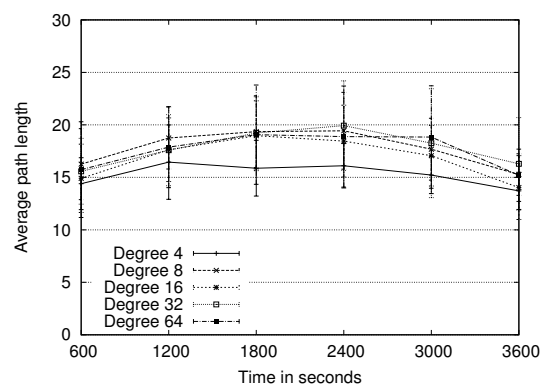


Figure 12. Average path length between peers.

peers always try to establish overlay links to hop-wise closer peers.

Figure 13 shows the 90th percentile value of the path length. Here also, the path length is measured as the number of IP hops covered by the packet. This value gives an acceptable statistical upper bound on the path length by excluding extreme cases. We can observe that the path length, for degrees above 4, is around 35 compared to the average path length of 18 seen in Figure 12. We conclude that including the values from the median to the 90th percentile yields a path inflation of 100% (i.e., paths are twice as long as the shortest ones) which is important but comparable to values measured at the IP layer [13].

We now evaluate the DHT efficiency. The only difference with the previous simulations is that now the peers do not send data packets but only storing and solving requests. The frequency of the storing requests generated in each peer is 1 every 30 seconds. The frequency of the solving requests generated in each peer is 1 every 5 seconds. We do not consider any redundancy parameters for now. Thus, a given pair is stored on one peer only. We observe the influence of the addressing tree degree of the overlay on

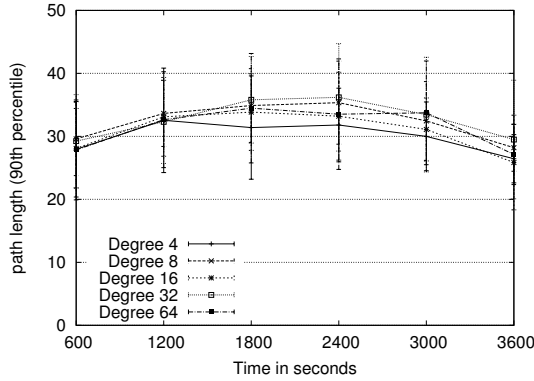


Figure 13. 90th percentile path length between peers.

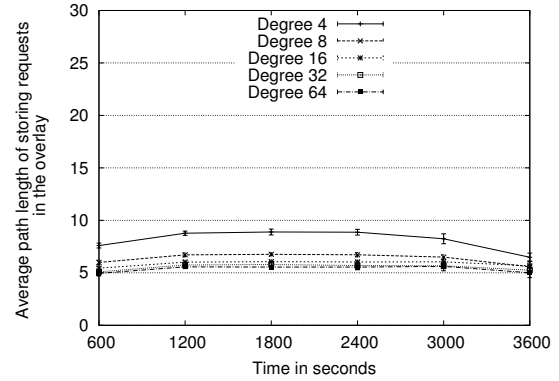


Figure 15. Average path length of the storing requests in the overlay.

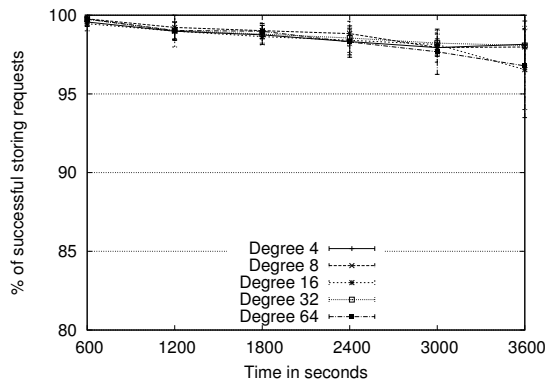


Figure 14. Percentage of successful storing requests.

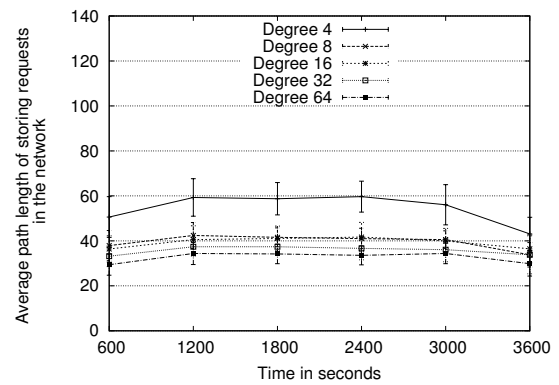


Figure 16. Average path length of the storing requests in the IP network.

the performances of the storing and the solving requests. More precisely we measure the rate of success as well as the average overlay path length of both storing and solving requests.

Figure 14 shows the percentage of successful storing requests over the simulation duration. We assume here that only one copy of a given pair is stored in the system. We can see that given the parameters of the simulation, the rate of success is very high despite the churn.

Figure 15 shows the average path length of the storing requests in the overlay network over the simulation duration. The number of peers to go through including the destination before storing a pair varies from 6 to 9 depending on the addressing tree degree. This number is decreasing when the degree is increasing with a diminishing return effect that can be seen starting at degree 16.

Figure 16 shows the average path length of the storing requests in the IP network over the simulation duration. We can see on this plot that the addressing tree degree has a greater impact on the number of IP hops than on the number of overlay hops. The number of hops are greater of course but also the variability of the values as well as the gaps

between the plots of the various degree parameter values are much higher. For a degree of 4 the average hop count is 60 whereas for a degree of 64 the average hop count is around 27. Given the results of Figure 15, we can deduce that the average IP hops between the peers varies from around 4.5 to 6.7 which is lower than the average path length of 7.9 measured in the IPv6 map. We can deduce that the peers which store the bindings are on average closer to the core of the network.

Figure 17 shows the percentage of successful solving requests over the simulation duration. As for the storing request, we can see that given the parameters of the simulation, the rate of success is very high despite the churn.

Figure 18 shows the average path length of the solving requests in the overlay network over the simulation duration. The number of peers to go through to reach the holder of the pair and including the return trip to the sender of the request varies roughly from 9 to 16 depending on the addressing tree degree. A degree of 4 yields a typical path length of 16, a degree of 8 reduces the path length to 12 and degree values above 8 all yield path lengths between 9 and 10. Thus the number of hops is decreasing when the degree is increasing

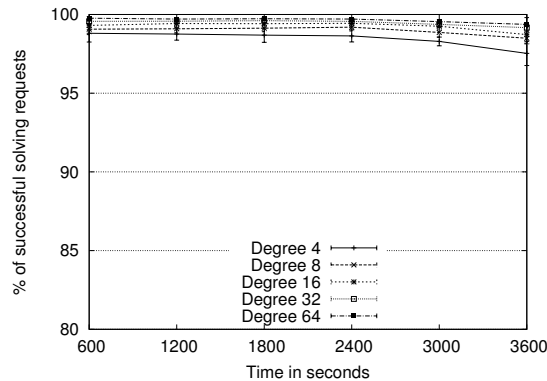


Figure 17. Percentage of successful solving requests.

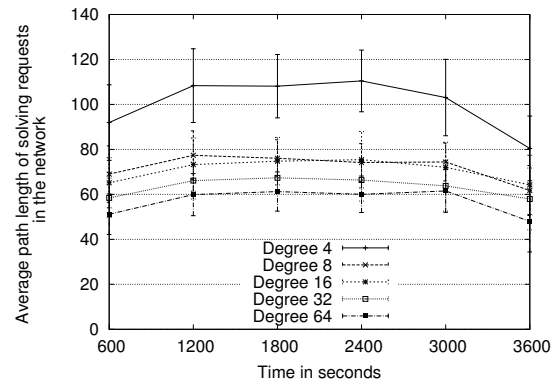


Figure 19. Average path length of the solving requests in the IP network.

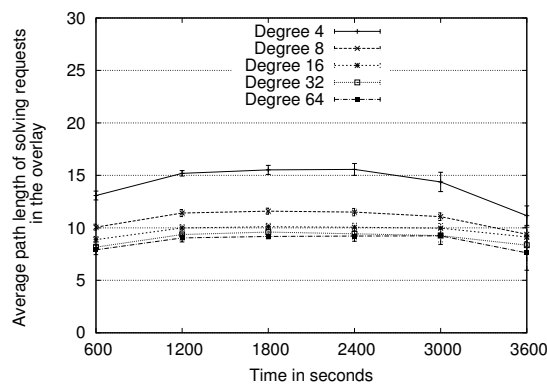


Figure 18. Average path length of the solving requests in the overlay.

with a diminishing return effect around degree 16, similar to the storing requests path lengths of Figure 15.

Figure 19 shows the average path length of the solving requests in the IP network over the simulation duration. We can also see on this plot that the addressing tree degree has a much greater impact on the number of IP hops than on the number of overlay hops. The gaps between the plots of the various degree parameter values are much higher than what was observed for the storing requests in Figure 16. If we except the plots corresponding to the degrees 8 and 16 that are very close to each other, the other plots have widely different path lengths ranging from roughly 110 for degree 4 to 60 for degree 64. Given the results of Figure 18, we can deduce that the average IP hops between the pairs of neighbor peers varies from around 6.6 to 6.9 which is a bit lower than the average path length of 7.9 in the IPv6 map. As a solving request takes a much longer path than a storing request, this explains the reduction in the variability of the number of IP hops between two neighbor peers.

We can conclude that given those simulation results, our overlay routing mechanism remains efficient under dynamics with a success rate above 90%. The average path lengths

in the overlay are typically between 15 and 20 IP network hops. Our DHT request shows encouraging performances whatever the degree chosen. The rate of success of both the storing and solving requests is above 95%. The average path lengths of the requests are also acceptable and show typical values for DHT systems.

In order to compare our DHT solution detailed in Section IV to previous existing schemes, we have implemented the addressing, routing and DHT mechanisms of CLOAK inside the PeerSim simulator. We have thus obtained comparative simulation results with Chord [14], Kademlia [15] and MSPastry [16] by using the same simulation parameters (e.g., simulation duration, peers' topology, peers' session lengths, etc). We have used an overlay network with a size remaining around 1000 nodes for 2 hours of simulated time. The churn rate varies from 10% to 60% over periods of 10 minutes (i.e., during the 10 minutes, x % of randomly selected peers will leave and be replaced by new ones). Each point on these plots, is the average of 10 runs and the standard deviation is provided.

Figure 20 shows the success ratio of the solving requests as a function of the churn rate. We can see that all DHT schemes perform similarly with a success ratio linearly decreasing with the churn rate. CLOAK has the best success ratio results, closely followed by MSPastry and Chord which have nearly the same values. Kademlia has the lowest success ratio results. As the plots for the storing requests are very similar to the solving ones, we do not show them to avoid redundancy.

Figure 21 shows the average path length measured by hop count of the solving requests as a function of the churn rate. Here again, the DHT schemes have the same behavior with a path length (in hops) slowly decreasing when the churn increases. MSPastry exhibits the shortest path lengths, closely followed by CLOAK. Kademlia has on average 1 more hop than MSPastry whatever the churn, while Chord has the longest path lengths, being on average 2 hops longer than MSPastry and CLOAK, although this difference tends

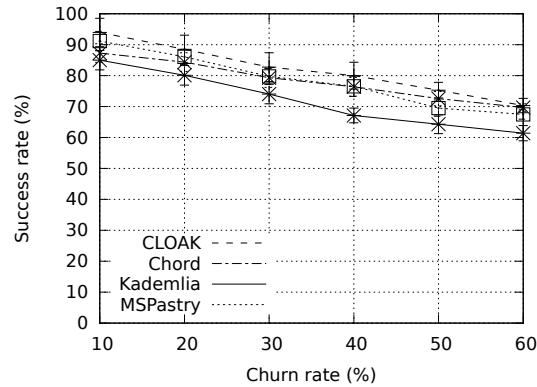


Figure 20. Comparison of the success ratio of the solving requests for various DHT vs churn rate.

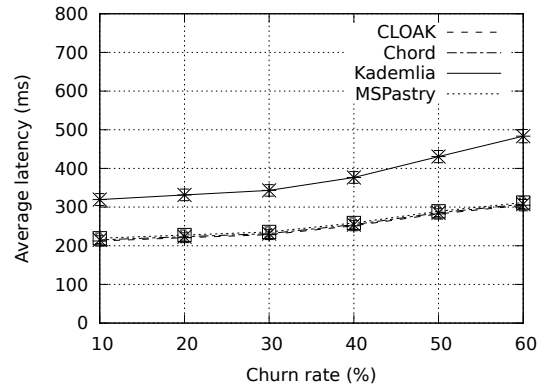


Figure 22. Comparison of the latency of the solving requests for various DHT vs churn rate.

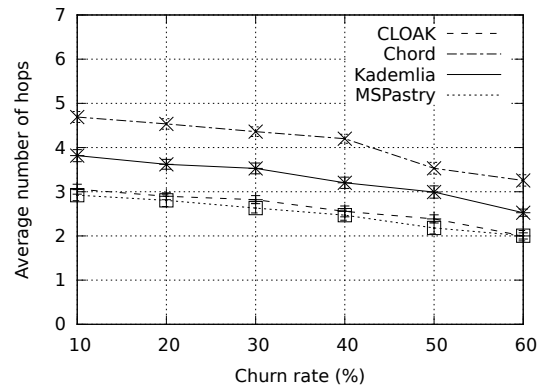


Figure 21. Comparison of the hop count of the solving requests for various DHT vs churn rate.

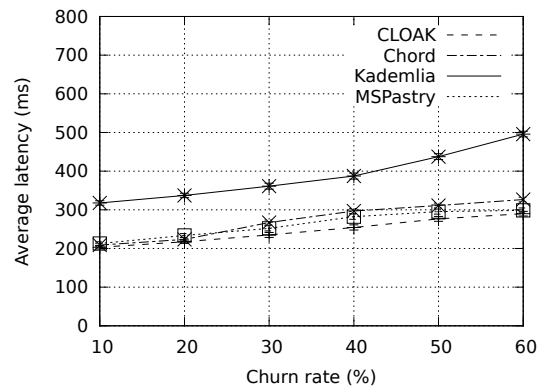


Figure 23. Comparison of the latency of the storing requests for various DHT vs churn rate.

to decrease when the churn is equal or above 40%. As for the success ratio, we do not show the plots for the hop count of the storing requests because they are very similar to the solving ones.

Figure 22 shows the average latency of the solving requests as a function of the churn rate. Indeed, as a path length measured in number of hops does not necessarily translate into a higher latency, we have measured the latter one to evaluate the time taken for the requests to complete. All the DHT schemes have nearly the same latency at any churn rate, excepted for Kademia which is typically 100 ms to 180 ms higher than the others depending on the churn rate. These results illustrate our point above that despite longer path lengths, Chord performs as well as MSPastry and CLOAK when the latency is observed.

Figure 23 shows the average latency of the storing requests as a function of the churn rate. Unlike the success ratio and hop count metrics above, here the plots of the storing requests are a bit different than the solving ones. Starting at 20% of churn, the plots of the latency of Chord,

MSPastry and CLOAK begin to separate. CLOAK has the lowest latency values, followed by MSPastry and Chord. However the gaps between these three plots are quite small, typically no more than 25 ms. As above for the solving latency, Kademia has a storing latency which is typically 100 ms to 170 ms higher than the others depending on the churn rate.

All these comparative results show that CLOAK performs as well as (and sometimes a little bit better than) Chord, MSPastry and Kademia which are the three popular DHT schemes that we have compared CLOAK to. These results also confirm our analysis presented in Section V. The key advantage of our solution is that peers can connect freely to any other peers they want, while in other DHT schemes such as Chord, peers must insert themselves in the DHT by connecting to other predetermined peers depending on their IP addresses. Another advantage is the cheap cost of building our DHT on top of our addressing and routing system. Using another DHT scheme would impose us to use two different routing schemes with the associated costs.

Our greedy hyperbolic routing scheme for the overlay and a key based routing scheme for the DHT. The simulation results encourage us to keep our own DHT scheme inside the CLOAK overlay.

VII. RELATED WORK

A. Transport Layer Mobility

Virtual connections, as we define them, can be considered as providing (among other benefits) transport layer connection mobility. Research on such transport layer connection mobility has mainly remained experimental up to now. Concerning the TCP connection management, several solutions have been proposed. TCP-Migrate [17], [18] developed at the Massachusetts Institute of Technology, provides a unified framework to support address changes and connectivity interruptions. TCP-Migrate provides mobile-aware applications with a set of system primitives for connectivity re-instantiation. TCP-Migrate enables applications to reduce their resource consumption during periods of disconnection and resume sessions upon reconnection. Rocks [19] developed at the University of Wisconsin, protects socket-based applications from network failures, such as link failures, IP address changes and extended periods of disconnection. Migratory TCP [20], developed at Rutgers University, is a transport layer protocol for building highly-available network services by means of transparent migration of the server endpoint of a live connection between cooperating servers that provide the same service. The origin and destination servers cooperate by transferring the connection state in order to accommodate the migrating connection. Finally, the Fault-Tolerant TCP [21], [22] developed at the University of Texas, ensures a faulty server to keep its TCP connections open until it either recovers or it is failed over to a backup. The failure and recovery of the server process are completely transparent to client processes. However, all these projects only deal with TCP re-connection. They do not provide a total virtualization of the communication and do not permit to switch both applications and/or devices from any communicating user at will. Furthermore, they are based on the domain name and IP address paradigm and do not provide the separation of the naming and addressing planes.

B. Name and Address Separation

Other solutions have been proposed with this separation in mind. However, they are typically placed below the transport layer and require modifications in the host or in the network infrastructure. The Host Identity Protocol (HIP) [23] for example, proposes a new namespace, the Host Identity namespace, and a new protocol layer named HIP, between the internetworking and transport layers. This solution requires the modification of the host stack. Similarly, the Locator/Identifier Separation Protocol (LISP) [24] is a network-based protocol that enables separation of IP addresses into

two new numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). No changes are required to either host protocol stacks or to the *core* of the Internet infrastructure. However LISP requires software changes in *edge* routers and cannot deal with host mobility. LISP can be incrementally deployed and offers traffic engineering and multi-homing benefits even when there are relatively few LISP-capable sites. The Shim6 protocol [25] is a layer 3 shim for providing locator agility below the transport protocols, so that multihoming can be provided for IPv6 with failover and load-sharing properties, without assuming that a multihomed site will have a provider-independent IPv6 address prefix announced in the global IPv6 routing table. Currently, this solution is restricted to the IPv6 network. The Internet Indirection Infrastructure (i3) by Stoica *et al.* [26] is an overlay-based indirection infrastructure that offers a rendezvous-based communication abstraction thus decoupling the act of sending from the act of receiving. Instead of sending a packet to a destination, each packet has an identifier which is used by the receiver to obtain the packet. However, i3 must be defined as a general overlay that everyone should use thus needing third party resources (such as the DNS infrastructure). With CLOAK, we try to enforce the principle that only the members of a given overlay have to share their resources. The Host Identity Indirection Infrastructure (Hi3) by Gurtov *et al.* [27] is a networking architecture for mobile hosts, derived from i3 and HIP. Although Hi3 provides efficient support for secure mobility and multihoming to Internet hosts, we do not adopt this infrastructure in order to avoid the issues of IP stack modifications (HIP) and third party resource requirements (i3). Data-Oriented Network Architecture (DONA) by Koponen *et al.* [28] proposes the use of permanent flat names coupled with name-based routing. Rather than use DNS servers, DONA relies on a new class of network entities called resolution handlers (RHs). As with i3, DONA needs third party resources provided by the infrastructure of RHs.

C. Distributed Hash Table

Concerning the DHT part of our solution, our proposal borrows some elements from well known DHTs. Our mapping mechanism for placing keys on the unit circle is similar to the one defined by Chord [14]. However, unlike Chord we do not place the peers themselves on this circle but inside the unit disk by using complex coordinates. Similarly to CAN [29], we use a multi-dimensional coordinate space, but instead of using a d -dimensional cartesian multi-torus, we use the 2-dimensional hyperbolic plane \mathbb{H}^2 . Our greedy routing scheme is based on a properly defined distance metric as done in Kademlia [15]. But unlike Kademlia which is based on the XOR metric, we use the hyperbolic distance defined for the Poincaré disk model of the hyperbolic plane. Another advantage of our greedy routing algorithm as opposed to prefix routing algorithms such as those developed

in Pastry [16], is that it does not rely on routing tables. Only the coordinates of the neighbors of a peer are needed to forward a message. This is highly scalable as the peers do not need to build and maintain routing tables. The intuition of using the hyperbolic plane as a virtual address space for our overlay and DHT systems comes from the work of Kleinberg [4]. However, we have defined a novel mapping function, whereas Kleinberg has suggested using CAN for implementing a DHT based on hyperbolic coordinates.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a new architecture called CLOAK designed for providing flexibility to Internet communications by using virtual connections set upon an overlay network. This architecture will be implemented as two protocols running on top of the transport protocols of the devices. The devices using the CLOAK middleware will freely interconnect with each other and thus will form a dynamic P2P overlay network. This overlay will enable the applications to maintain their communications even if some transport layer connections are subject to failures. The middleware will transparently restore the transport connections without killing the connections of the applications. The architecture, by giving identifiers to users and devices, will provide flexibility, security and mobility to applications despite the IP address changes suffered by the devices.

We have also shown that given an appropriate mapping function, it is easy to setup and maintain a consistent DHT structure upon such an overlay. Our theoretical analysis has shown that our DHT proposal is scalable with performances similar to other existing DHTs. We have implemented the overlay addressing and routing part as well as the DHT part of our middleware in our discrete event *nem* simulator as well as in the *PeerSim* simulator and the results are encouraging. Our simulation results have demonstrated that the success rate of the routing procedure, as well as the success rate of the storing and solving requests are typical of such systems. Measurements of path lengths and latencies also confirm the proper behavior of our solution compared to prior ones.

Our future work will be aimed at defining the CLOAK API, implementing the middleware as a library, modifying a relevant test application (such as a chat or video streaming application) and testing it on a virtualized platform for studying the impact of transport layer connection pipelining created by the P2P overlay network.

REFERENCES

- [1] T. Tiendrebeogo, D. Magoni, and O. Sié, "Virtual internet connections over dynamic peer-to-peer overlay networks," in *Proceedings of the 3rd International Conference on Evolving Internet*, 2011, pp. 58–65.
- [2] C. Cassagnes, D. Bromberg, and D. Magoni, "An overlay architecture for achieving total flexibility in internet communications," in *Proceedings of the 8th International Conference on Advanced Information Technologies for Management*, 2010, pp. 39–60.
- [3] C. Cassagnes, T. Tiendrebeogo, D. Bromberg, and D. Magoni, "Overlay addressing and routing system based on hyperbolic geometry," in *Proceedings of the 16th IEEE Symposium on Computers and Communications*, 2011, pp. 294–301.
- [4] R. Kleinberg, "Geographic routing using hyperbolic space," in *Proceedings of the 26th IEEE INFOCOM*, 2007, pp. 1902–1909.
- [5] Rosenberg, Schulzrinne, Camarillo, Johnston, and P. et al., "SIP: Session initiation protocol," Internet Engineering Task Force, Request For Comments 3261, June 2002.
- [6] ITU-T, "H323: Packet-based multimedia communications systems," ITU-T, Recommendation, December 2009.
- [7] J. Postel and J. Reynolds, "File transfer protocol (FTP)," *Internet Engineering Task Force*, Request For Comments 959, 1985.
- [8] G. Wright and R. Stevens, *TCP/IP Illustrated, Volume 2: The Implementation*. Addison-Wesley, 1995.
- [9] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials*, vol. 7, pp. 72–93, 2005.
- [10] A.-L. Barabási and E. Bonabeau, "Scale-free networks," *Scientific American*, vol. 288, pp. 60–69, 2003.
- [11] R. Cohen and S. Havlin, "Scale-free networks are ultrasmall," *Phys. Rev. Lett.*, vol. 90, no. 5, p. 058701, Feb 2003.
- [12] D. Magoni, "Network topology analysis and internet modelling with nem," *International Journal of Computers and Applications*, vol. 27, no. 4, pp. 252–259, 2005.
- [13] D. Magoni and M. Hoerd, "Internet core topology mapping and analysis," *Computer Communications*, vol. 28, no. 5, pp. 494–506, 2005.
- [14] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM SIGCOMM Conference*, 2001, pp. 149–160.
- [15] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS)*. Springer-Verlag, 2002, pp. 53–65.
- [16] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, 2001, pp. 329–350.
- [17] A. C. Snoeren, H. Balakrishnan, and M. F. Kaashoek, "Reconsidering IP mobility," in *Proceedings of the 8th HotOS*, 2001, pp. 41–46.

- [18] A. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility," in *Proceedings of the 6th ACM MobiCom*, 2000, pp. 155–166.
- [19] V. Zandy and B. Miller, "Reliable network connections," in *Proceedings of the 8th ACM MobiCom*, 2002, pp. 95–106.
- [20] F. Sultan, K. Srinivasan, D. Iyer, and L. Iftode, "Migratory TCP: Connection migration for service continuity in the internet," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*, 2002, pp. 469–470.
- [21] D. Zagorodnov, K. Marzullo, and T. Bressoud, "Engineering fault tolerant TCP/IP services using FT-TCP," in *Proceedings of the IEEE International Conference on Dependable Systems and Networks*, 2003, pp. 393–402.
- [22] T. Bressoud, A. El-Khashab, K. Marzullo, and D. Zagorodnov, "Wrapping server-side TCP to mask connection failures," in *Proceedings of the 20th IEEE INFOCOM*, 2001, pp. 329–338.
- [23] P. Nikander, A. Gurtov, and T. Henderson, "Host identity protocol (hip): Connectivity, mobility, multi-homing, security, and privacy over ipv4 and ipv6 networks," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 186–204, 2010.
- [24] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/identifier separation protocol," *Internet Engineering Task Force*, Internet Draft, July 2011.
- [25] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for IPv6," *Internet Engineering Task Force*, Request For Comments 5533, June 2009.
- [26] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," in *Proceedings of ACM SIGCOMM'02*, 2002.
- [27] A. Gurtov, D. Korzun, A. Lukyanenko, and P. Nikander, "Hi3: An efficient and secure networking architecture for mobile hosts," *Computer Communications*, vol. 31, pp. 2457–2467, 2008.
- [28] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '07, 2007, pp. 181–192.
- [29] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the ACM SIGCOMM Conference*, 2001, pp. 161–172.

Quality Analysis of a Chaotic Proven Keyed Hash Function 26

Jacques M. Bahi, Jean-François Couchot, and Christophe Guyeux
University of Franche-Comté, FEMTO-ST Institute
Belfort, France

Email: {jacques.bahi, jean-francois.couchot, christophe.guyeux}@univ-fcomte.fr

Abstract—Hash functions are cryptographic tools, which are notably involved in integrity checking and password storage. They are of primary importance to improve the security of exchanges through the Internet. However, as security flaws have been recently identified in the current standard in this domain, new ways to hash digital data must be investigated. In this document an original keyed hash function is evaluated. It is based on asynchronous iterations leading to functions that have been proven to be chaotic. It thus possesses various topological properties as uniformity and sensibility to its initial condition. These properties make our hash function satisfies established security requirements in this field. This claim is qualitatively proven and experimentally verified in this research work, among other things by realizing a large number of simulations.

Keywords—Keyed Hash Function; Internet Security; Mathematical Theory of Chaos; Topology.

I. INTRODUCTION

The security and the privacy of data exchanged through the Internet are guaranteed by protocols that make an adequate use of a few cryptographic tools as secure pseudorandom number generators or hash functions. Hash functions are applications that map words of any lengths to words of fixed lengths (often 256 or 512 bits). These hash functions allow, for instance, to store passwords in a secure manner or to check whether a download has occurred without any error. They be designed to depend from a given parameter, called a key. According to their field of application, the requirements an hash function has to satisfy can change. They need at least to be very fast, so that the diffusion of the digest into the set of hash values occurs (whatever the bias into the inputted message), and so that a link between a message and its digest is impossible to establish in practice (confusion). The possibility to use a key or to distribute the computation on numerous threads must often be offered in several applications. Finally, in the computer security field, stringent complexity properties have to be proven, namely the collision, preimage, and second-preimage resistances, the unpredictability, and the pseudorandomness properties. Each of the latter one have a rigorous formulation in terms of polynomial indistinguishability.

Several hash functions have been proposed as candidates to be standards in computer science. Such standards are designed by the scientific community and selected, after peer studies, by administrations as the NIST one (National Institute for Standards and Technologies of the US government). SHA-1 is probably the most widely used hash function. It is present in a large panel of security applications and protocols through the Internet.

However, in 2004, MD5 and SHA-0 have been broken. An attack over SHA-1 has been achieved with only 2^{69} operations (CRYPTO-2005), that is, 2,000 times faster than a brute force

attack (that requires 2^{80} operations). Even if 2^{69} operations still remain impossible to realize on common computers, such a result, based on a previous attack on SHA-0, is a very important one: as the SHA-2 variants are algorithmically close to SHA-1 and eventually produce message digests on principles similar to the MD4 and MD5 message digest algorithms, a new hash standard based on original approaches is then eagerly awaited. This is why a SHA-3 contest has been launched these last few years, to find a new, more secure standard for hash functions. So new original hash functions, or improvements for existing ones, must be found.

In this context, we have proposed a new hash function in [1], [2]. Being designed by using discrete dynamical systems, and taking benefits from various established topological properties, this new family of hash functions is thus based on a completely different approach. Among other things, in our proposal, an ingredient of chaos is added to existing hash functions, in order to reinforce their properties. Introducing chaos into the design of hash functions has been already addressed in [3], [4], [5], [6]. These methods usually transform the initial message into its padded fixed length version and then translate it into a real number. Next, with a chosen chaotic map (some chaotic functions of real variables like logistic, tent, or Arnold's cat maps, for instance [7]), methods set the initial algorithm parameters according to the secret key and start iterations. Methods are then left to extract some bits from the iterations results and to juxtapose them to get the hash value. It is then supposed that the final hash function preserves the properties of chaos. However, the idea of chaotic hash functions has been controversially discussed in the community [8], [9]. Moreover, even if these algorithms are themselves proven to be chaotic, their implementations on finite machines can result into the loss of chaos property. Among other things, the main reason is that chaotic functions (embedded in these researches) only manipulate real numbers, which do not exist in a computer. In [2], the hash function we have proposed does not simply integrate chaotic maps into algorithms hoping that the result remains chaotic; we have conceived an algorithm and have mathematically proven that it is chaotic. To do both, our theory and our implementation are based on finite integer domains and finite states iterations, where only one randomly chosen element is modified at each step. This iteration mode is further referred to as asynchronous mode.

These studies lead to the conclusion that the chaos of asynchronous iterations is very intense [10]. As this mode only manipulates binary digits or integers, we have shown that truly chaotic computer programs can be produced. They can thus be applied to pseudorandom number generators [11] and to a complete class of information hiding schemes [12],

for instance. In this paper, the complete chaotic behavior of asynchronous iterations is capitalized to produce a truly chaotic keyed hash function.

This research work is an improvement of a previous article accepted at the Third International Conference on Evolving Internet, INTERNET11 (June 19-24, 2011, Luxembourg) [1]. Compared to this research work, the proposed hash function (Section III) has been completely rethought. It appears now more as a post-treatment on existing hash functions, to improve their security (Sections III, IV), than as a hash function designed from scratch. Moreover, the second-preimage resistance has been proven in Section IV-B and the strict avalanche criterion has been statistically studied (Section V-C). All these improvements lead to obviously better scores for the proposed hash functions, when experimentally evaluating its security.

The remainder of this research work is organized in the following way. In Section II, basic notions concerning asynchronous iterations and Devaney's chaos are recalled. Our keyed hash function is presented in Section III. Performance analyses are presented in the next two sections: in the first one a qualitative evaluation of this function is outlined, whereas in the second one it is evaluated experimentally. This research work ends by a conclusion section, in which our contribution is summarized and intended future work is mentioned.

II. BACKGROUND SECTION

In this section, we first give definitions of Secure Keyed One-Way Hash Functions and of the Strict Avalanche Criterion (SAC), which is a property that such a function has to verify. Next we give some recalls on Boolean discrete dynamical systems and link them with topological chaos. Finally, we establish relations between the algorithm properties inherited from topological results and the requirements of Secure Keyed One-Way Hash Function.

A. Secure Keyed One-Way Hash Function

Definition 1 (Secure Keyed One-Way Hash Function [13]) Let Γ and Σ be two alphabets, let $k \in K$ be a key in a given key space, let l be a natural number, which is the length of the output message, and let $h : K \times \Gamma^+ \rightarrow \Sigma^l$ be a function that associates a message in Σ^l for each pair of key, word in $K \times \Gamma^+$. The set of all functions h is partitioned into classes of functions $\{h_k : k \in K\}$ indexed by a key k and such that $h_k : \Gamma^+ \rightarrow \Sigma^l$ is defined by $h_k(m) = h(k, m)$, i.e., h_k generates a message digest of length l .

A class $\{h_k : k \in K\}$ is a Secure Keyed One-Way Hash Function if it satisfies the following properties:

- 1) the function h_k is keyed one-way. That is,
 - a) Given k and m , it is easy to compute $h_k(m)$.
 - b) Without the full knowledge of k , it is
 - difficult to find m when $h_k(m)$ is given; this property is referred to as preimage resistance;
 - difficult to find $h_k(m)$ when only m is given.
- 2) The function h_k is the keyed collision resistant, that is, without the knowledge of k it is difficult to find two distinct messages m and m' s.t. $h_k(m) = h_k(m')$. A weaker version of this property is the second preimage resistance, which is established if for a given m it is difficult to find another message m' , $m \neq m'$, such that $h_k(m) = h_k(m')$.

- 3) Images of function h_k have to be uniformly distributed in Σ^l in order to counter statistical attacks. 27
- 4) Length l of the produced image has to be larger than 128 bits in order to counter birthday attacks [14].
- 5) Key space size has to be sufficiently large in order to counter exhaustive key search.

Finally, hash functions have to verify the *strict avalanche criterion* defined as follows:

Definition 2 (Strict Avalanche Criterion [15]) Let x and \bar{x}^i , two n -bit, binary vectors, such that x and \bar{x}^i differ only in bit i , $1 \leq i \leq n$. Let f be the cryptographic transformation (hash function applied on vector of bits for instance). Let \oplus be the exclusive or operator. The f function meets the strict avalanche criterion if and only if the following property is established;

$$\forall n. \forall i, j. 1 \leq i \leq n \wedge 1 \leq j \leq m \Rightarrow P\left((f(x) \oplus f(\bar{x}^i))_j = 1\right) = 1/2$$

This means that for any length message, each bit of the digest is independent of modifying one bit in the original message. In other words, a difference of one bit between two given medias has to lead to completely different digests.

B. Boolean Discrete Dynamical Systems

Let us first discuss the domain of iterated functions. As far as we know, no result rules that the chaotic behavior of a function that has been theoretically proven on \mathbb{R} remains valid on the floating-point numbers, which is the implementation domain. Thus, to avoid the loss of chaos this research work presents an alternative, namely to iterate Boolean maps: results that are theoretically obtained in that domain are preserved during implementations. This section recalls facts concerning Boolean discrete-time dynamical Systems (BS) that are sufficient to understand the background of our approach.

Let us denote by $\llbracket a; b \rrbracket$ the interval of integers: $\{a, a+1, \dots, b\}$, where $a \leq b$. Let n be a positive integer. A Boolean discrete-time system is a discrete dynamical system defined from a Boolean map $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ s.t.

$$x = (x_1, \dots, x_n) \mapsto f(x) = (f_1(x), \dots, f_n(x)),$$

and an iteration scheme: parallel, asynchronous... With the parallel iteration scheme, the dynamics of the system are described by $x^{t+1} = f(x^t)$ where $x^0 \in \mathbb{B}^n$. Let thus $F_f : \llbracket 1; n \rrbracket \times \mathbb{B}^n$ to \mathbb{B}^n be defined by

$$F_f(i, x) = (x_1, \dots, x_{i-1}, f_i(x), x_{i+1}, \dots, x_n),$$

with the *asynchronous* scheme, the dynamics of the system are described by $x^{t+1} = F_f(s^t, x^t)$ where $x^0 \in \mathbb{B}^n$ and s is a strategy, i.e., a sequence in $\llbracket 1; n \rrbracket^{\mathbb{N}}$. Notice that this scheme only modifies one element at each iteration.

Let G_f be the map from $\mathcal{X} = \llbracket 1; n \rrbracket^{\mathbb{N}} \times \mathbb{B}^n$ to itself s.t.

$$G_f(s, x) = (\sigma(s), F_f(s^0, x)),$$

where $\sigma(s)^t = s^{t+1}$ for all t in \mathbb{N} . Notice that the parallel iteration of G_f from an initial point $X^0 = (s, x^0)$ describes the "same dynamics" as the asynchronous iteration of f induced by the initial point x^0 and the strategy s .

The state-vector $x^t = (x_1^t, \dots, x_n^t) \in \mathbb{B}^n$ of the system at discrete time t (also said at *iteration* t) is further denoted as the *configuration* of the system at time t .

In what follows, the dynamics of the system is particularized with the negation function $\neg : \mathbb{B}^n \rightarrow \mathbb{B}^n$ such that $\neg(x) = (\bar{x}_1, \dots, \bar{x}_n)$ where \bar{x}_i is the negation of x_i . We thus have the function F_{\neg} that is defined by:

$$F_{\neg} : \llbracket 1; n \rrbracket \times \mathbb{B}^n \rightarrow \mathbb{B}^n$$

$$F_{\neg}(s, x)_j = \begin{cases} \bar{x}_j & \text{if } j = s \\ x_j & \text{otherwise.} \end{cases}$$

With such a notation, configurations are defined for times $t = 0, 1, 2, \dots$ by:

$$\begin{cases} x^0 \in \mathbb{B}^n \text{ and} \\ x^{t+1} = F_{\neg}(S^t, x^t) \end{cases} \quad (1)$$

In the space $\mathcal{X} = \llbracket 1; n \rrbracket^{\mathbb{N}} \times \mathbb{B}^n$ we define the distance between two points $X = (S, E), Y = (\check{S}, \check{E}) \in \mathcal{X}$ by

$$d(X, Y) = d_e(E, \check{E}) + d_s(S, \check{S}), \text{ where}$$

$$d_e(E, \check{E}) = \sum_{k=1}^n \delta(E_k, \check{E}_k), \text{ and}$$

$$d_s(S, \check{S}) = \frac{9}{n} \sum_{k=1}^{\infty} \frac{|S^k - \check{S}^k|}{10^k}.$$

If the floor value $\lfloor d(X, Y) \rfloor$ is equal to j , then the systems E, \check{E} differ in j cells. In addition, $d(X, Y) - \lfloor d(X, Y) \rfloor$ is a measure of the differences between strategies S and \check{S} . More precisely, this floating part is less than 10^{-k} if and only if the first k terms of the two strategies are equal. Moreover, if the k^{th} digit is nonzero, then the k^{th} terms of the two strategies are different.

In his PhD thesis [10], Guyeux has already proven that:

- The function G_f is *continuous* on the metric space (\mathcal{X}, d) .
- The parallel iterations of G_{\neg} are *regular*: periodic points of G_{\neg} are dense in \mathcal{X} .
- The function G_{\neg} is *topologically transitive*: for all $X, Y \in \mathcal{X}$, and for all open balls B_X and B_Y centered in X and Y respectively, there exist $X' \in B_X$ and $t \in \mathbb{N}$ such that $G_{\neg}^t(X') \in B_Y$.
- The function G_{\neg} has *sensitive dependence on initial conditions*: there exists $\delta > 0$ such that for any $X \in \mathcal{X}$ and any open ball B_X , there exist $X' \in B_X$ and $t \in \mathbb{N}$ such that $d(G_{\neg}^t(X), G_{\neg}^t(X')) > \delta$.

To put it differently, a system is sensitive to initial conditions if any point contains, in any neighborhood, another point with a completely different future trajectory. Topological transitivity is established when, for any element, any neighborhood of its future evolution eventually overlaps with any other open set. On the contrary, a dense set of periodic points is an element of regularity that a chaotic dynamical system has to exhibit.

We have previously established that the three conditions for Devaney's chaos hold for asynchronous iterations. They thus behave chaotically, as it is defined in the mathematical theory of chaos [16], [17]. In other words, quoting Devaney in [16], a chaotic dynamical system "is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or simplified into two subsystems, which do not interact because of topological transitivity. And in the midst of this random behavior, we nevertheless have an element of regularity".

Intuitively, the topological transitivity and the sensitivity on initial conditions respectively address the preimage resistance and the avalanche criteria. Section IV formalizes this intuition.

The next section presents our hash function that is based on asynchronous iterations.

III. CHAOS-BASED KEYED HASH FUNCTION ALGORITHM

The hash value is obtained as the last configuration resulting from iterations of G_{\neg} . We then have to define the pair $X^0 = ((S^t)^{t \in \mathbb{N}}, x^0)$, i.e., the strategy S and the initial configuration x^0 .

A. Computing x^0

The first step of the algorithm is to transform the message in a normalized $n = 256$ bits sequence x^0 . Notice that this size n of the digest can be changed, *mutatis mutandis*, if needed. Here, this first step is close to the pre-treatment of the SHA-1 hash function, but it can easily be replaced by any other compression method.

To illustrate this step, we take an example, our original text is: "The original text".

Each character of this string is replaced by its ASCII code (on 7 bits). Following the SHA-1 algorithm, first we append the character "1" to this string, which is then

```
10101001 10100011 00101010 00001101 11111100
10110100 11100111 11010011 10111011 00001110
11000100 00011101 00110010 11111000 11101001.
```

Next we append the block 1111000, which is the binary value of this string length (120) and let R be the result. Finally another "1" is appended to R if and only if the resulting length is an even number.

```
10101001 10100011 00101010 00001101 11111100
10110100 11100111 11010011 10111011 00001110
11000100 00011101 00110010 11111000 11101001
11111000.
```

The whole string is copied, but in the opposite direction:

```
10101001 10100011 00101010 00001101 11111100
10110100 11100111 11010011 10111011 00001110
11000100 00011101 00110010 11111000 11101001
11110000 00111110 01011100 01111101 00110010
11100000 10001101 11000011 01110111 00101111
10011100 10110100 11111110 11000001 01010011
00010110 010101.
```

The string whose length is a multiple of 512 is obtained, by duplicating the string obtained above a sufficient number of times and truncating it at the next multiple of 512. This string is further denoted by D . Finally, we split our obtained string into two blocks of 256 bits and apply to them the exclusive-or (further denoted as XOR) function, from the first two blocks to the last one. It results a 256 bits sequence, that is in our example:

```
00001111 00101111 10000010 00111010 00001110
01100111 01111000 10011101 01010111 00110101
11010100 01101001 11111001 00011011 01001110
00110000 11000111 00101101 10001001 11111001
01100010 10111010 11001110 10101011 10010001
11101110 01100111 00000101 11000100 00011111
01001111 00001100.
```

The configuration x^0 is the result of this pre-treatment and is a sequence of $n = 256$ bits. Notice that many distinct

texts lead to the same string x^v . The algorithm detailed in [1] always appends “1” to the string R . However such an approach suffered from generating the same x^0 when R ’s length is 128. In that case the size of its reverse is again 128 bits leading a message of length 256. When we duplicate the message, we obtain a message of length 512 composed of two equal messages. The resulting XOR function is thus 0 and this improvement consequently allows us to avoid this drawback.

Let us build now the strategy $(S^t)^{t \in \mathbb{N}}$ that depends on the original message and on a given key.

B. Computing $(S^t)^{t \in \mathbb{N}}$

To obtain the strategy S , the chaotic proven pseudorandom number generator detailed in [18] is used. The seed of this PRNG is computed as follows: first the ASCII code (on 7 bits again) of the key is duplicated enough and truncated to the length of D . A XOR between D and this chain gives the seed of the PRNG, that is left to generate a finite sequence of natural numbers S^t in $\llbracket 1, n \rrbracket$ whose length is $2n$.

C. Computing the digest

To design the digest, asynchronous iterations of G_{\neg} are realized with initial state $X^0 = ((S^t)^{t \in \mathbb{N}}, x^0)$ as defined above. The result of these iterations is a $n = 256$ bits vector. Its components are taken 4 per 4 bits and translated into hexadecimal numbers, to obtain the hash value:

```
AF71542C90F450F6AE3F649A0784E6B1
6B788258E87654B4D6353A2172838032.
```

As a comparison if we replace “*The original text*” by “*the original text*”, the hash function returns:

```
BAD8789AD6924B6460F8E7686A24A422
8486DC8FDCAE15F1F681B91311426056.
```

We then investigate the qualitative properties of this algorithm.

IV. QUALITY ANALYSIS

We show in this section that, as a consequence of recalled theoretical results, this hash function tends to verify desired informal properties of a secure keyed one-way hash function.

A. The Strict Avalanche Criterion

In our opinion, this criterion is implied by the topological properties of sensitive dependence to the initial conditions, expansivity, and Lyapunov exponent. These notions are recalled below.

First, a function f has a constant of expansivity equal to ε if an arbitrarily small error on any initial condition is *always* magnified till ε . In our iteration context and more formally, the function G_{\neg} verifies the *expansivity* property if there exists some constant $\varepsilon > 0$ such that for any X and Y in \mathcal{X} , $X \neq Y$, we can find a $k \in \mathbb{N}$ s.t. $d(G_{\neg}^k(X), G_{\neg}^k(Y)) \geq \varepsilon$. We have proven in [19] that, (\mathcal{X}, G_{\neg}) is an expansive chaotic system. Its constant of expansivity is equal to 1.

Next, some dynamical systems are highly sensitive to small fluctuations into their initial conditions. The constants of sensibility and expansivity have been historically defined to illustrate this fact. However, in some cases, these variations can become enormous, can grow in an exponential manner in

a few iterations, and neither sensitivity nor expansivity are able to measure such a situation. This is why Alexander Lyapunov²⁹ has proposed a new notion able to evaluate the amplification speed of these fluctuations we now recall:

Definition 3 (Lyapunov Exponent) *Let be given an iterative system $x^0 \in \mathcal{X}$ and $x^{t+1} = f(x^t)$. Its Lyapunov exponent is defined by:*

$$\lim_{t \rightarrow +\infty} \frac{1}{t} \sum_{i=1}^t \ln |f'(x^{i-1})|$$

By using a topological semi-conjugation between \mathcal{X} and \mathbb{R} , we have proven in [10] that, for almost all X^0 , the Lyapunov exponent of asynchronous iterations G_{\neg} with X^0 as initial condition is equal to $\ln(n)$.

We can now justify why, in our opinion, the topological properties of the proposed hash function lead to the avalanche effect. Indeed, due to the sensitive dependence to the initial condition, two close media can possibly lead to significantly different digests. The expansivity property implies that these similar medias mostly lead to very different hash values. Finally, a Lyapunov exponent greater than 1 leads to the fact that these two close media will always end up by having very different digests.

B. Preimage Resistance

1) *Topological Justifications:* Let us now discuss about the preimage resistance of our keyed hash function denoted by h . As recalled previously, an adversary given a target image D should not be able to find a preimage M such that $h(M) = D$. One reason (among many) why this property is important is that, on most computer systems, users passwords are stored as the cryptographic hash of the password instead of just the plain-text password. Thus an attacker who gains access to the password file cannot use it to then gain access to the system, unless it is able to invert target message digest of the hash function.

We now explain why, topologically speaking, our hash function is resistant to preimage attacks. Let m be the message to hash, (S, x^0) its normalized version (*i.e.*, the initial state of our iteration scheme), and $M = h(m)$ the digest of m by using our method. So iterations with initial condition (S, M) and iterate function G_{\neg} have x^0 as final state. Thus it is impossible to invert the hash process with a view to obtain the normalized message by using the digest. Such an attempt is equivalent to try to forecast the future evolution of asynchronous iterations of the \neg function by only using a partial knowledge of its initial condition. Indeed, as M is known but not S , the attacker has an uncertainty on the initial condition. He/she only knows that this value is into an open ball of radius 1 centered at the point M , and the number of terms of such a ball is infinite.

With such an uncertainty on the initial condition, and due to the numerous chaos properties possessed by our algorithm (as stated in the previous Section), this prediction is impossible. Furthermore, due to the transitivity property, it is possible to reach all of the normalized medias, when starting to iterate into this open ball. These qualitative explanations can be formulated more rigorously, by the proofs given in the next section.

2) *Proofs of the Second-Preimage Resistance:* We will focus now on a rigorous proof of the second-preimage resistance: an adversary given a message m should not be able to find another message m' such that $m \neq m'$ and $h(m) = h(m')$.

More precisely, we will show that a more general instance of the proposed post-treatment described below preserves this character for a given hash function.

Let

- k_1, k_2, n , all in \mathbb{N}^* , where k_1 is the size of the key, k_2 is the size of the seed, and n is the size of the hash value,
- $h : (k, m) \in \mathbb{B}^{k_1} \times \mathbb{B}^* \mapsto h(k, m) \in \mathbb{B}^n$ a keyed hash function,
- $S : k \in \mathbb{B}^{k_2} \mapsto S(k) \in \llbracket 1, n \rrbracket$ a cryptographically secure pseudorandom number generator,
- $\mathcal{K} = \mathbb{B}^{k_1} \times \mathbb{B}^{k_2} \times \mathbb{N}$ called the *key space*,
- and $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ a bijective map.

We define the keyed hash function $\mathcal{H} : \mathcal{K} \times \mathbb{B}^* \rightarrow \mathbb{B}^n$ by the following procedure

Inputs: $K = (K_1, K_2, N) \in \mathcal{K}$
 $m \in \mathbb{B}^*$
Runs: $X = h(K_1, m)$
for $i = 1, \dots, N$:
 $X = G_f(S^i(K_2), X)$
return X

where K_1 is the key of the inputted hash function, K_2 the seed of the strategy used in the post-treatment iterations, where N is for the size of this strategy. We have the following result.

Theorem 1 *If h satisfies the second-preimage resistance property, then it is the case for \mathcal{H} too.*

To achieve the proof, we introduce the two following lemmas.

Lemma 1 *If $f : \mathbb{B}^n \rightarrow \mathbb{B}^n$ is bijective, then for any $S \in \llbracket 1, n \rrbracket$, the map $G_{f,S} : x \in \mathbb{B}^n \rightarrow G_f([S, 1, \dots, 1], x)_2 \in \mathbb{B}^n$ is bijective too where $G_f(_, _)_2$ is the second term of the pair $G_f(_, _)$.*

Proof: Since \mathbb{B}^n is a finite set, it is sufficient to prove that $G_{f,S}$ is surjective. Let $y = (y_1, \dots, y_n) \in \mathbb{B}^n$ and $S \in \llbracket 1, n \rrbracket$. Thus $G_{f,S}((y_1, \dots, y_{S-1}, f^{-1}(y_S), y_{S+1}, \dots, y_n))_2 = G_f([S, 1, \dots, 1], (y_1, \dots, y_{S-1}, f^{-1}(y_S), y_{S+1}, \dots, y_n))_2 = ([1, \dots, 1], (y_1, \dots, y_{S-1}, y_S, y_{S+1}, \dots, y_n))_2 = (y_1, \dots, y_{S-1}, y_S, y_{S+1}, \dots, y_n) = y$. So $G_{f,S}$ is a surjective map between two finite sets and thus bijective. ■

Lemma 2 *Let $S \in \llbracket 1, n \rrbracket^{\mathbb{N}}$ and $N \in \mathbb{N}^*$. If f is bijective, then $G_{f,S,N} : x \in \mathbb{B}^n \mapsto G_f^N(S, x)_2 \in \mathbb{B}^n$ is bijective too.*

Proof: Indeed, $G_{s,f,n} = G_{f,S^n} \circ \dots \circ G_{f,S^0}$ is bijective as a composition of bijective maps (as stated in Lemma 1). ■

We are now able to prove theorem 1.

Proof: Let $m, k \in \mathbb{B}^* \times \mathcal{K}$. If a message $m' \in \mathbb{B}^*$ can be found such that $\mathcal{H}(k, m) = \mathcal{H}(k, m')$, then, according to Lemma 2, $h(k_1, m) = h(k_1, m')$: a second-preimage for h has thus been found. ■

C. Algorithm Complexity

In this section the complexity of the above hash function is evaluated for a size l of the media (in bits).

Theorem 2 *Let l be the size of the message to hash and n be the size of its hash value. The algorithm detailed along these*

lines requires $\mathcal{O}(l) + \mathcal{O}(n^e)$ elementary operations to produce the hash value.

Proof: In the x^0 computation stage only linear operations over binary tables are achieved. More precisely it first executes one ASCII translation yielding a message of length $7l$, a length computation that increases the message length of $\log_2(7l)$. One bit is possibly added. Thus a reversed copy that leads to a message of length l' that is $14l + 2 + 2\log_2(7l)$. The number of duplication steps to get a message whose length is greater than a multiple of $2n$ is formally given by

$$\min_{k \geq 1} \{k \mid \exists p. p \geq 1 \wedge k \times l' \geq 2np\} \quad (2)$$

This number is bounded by

$$k' = \max\{1, n\}.$$

If $14l + 2 + 2\log_2(7l)$ is greater than $2n$ it is sufficient to duplicate the message once. Otherwise, $\lfloor 1 + \frac{2n}{14l + 2 + 2\log_2(7l)} \rfloor$ is greater than $\frac{2n}{14l + 2 + 2\log_2(7l)}$ and thus $l' \times \frac{2n}{14l + 2 + 2\log_2(7l)}$ is greater than $2n$ and there exists a p ($p = 1$) such that $k \times l' \geq 2np$. Thus the minimum of the set given in Eq.(2) is less than $\lfloor 1 + \frac{2n}{14l + 2 + 2\log_2(7l)} \rfloor$, which is less than n .

To sum up, the initialization of x^0 requires at most $k' + l'$ elementary operations.

Let us now detail the S computation step. The number of elementary operations to provide the seed is bounded by $k' + l'$. Next, the embedded PRNG [18], that combines the XORShift, xor128, and XORWow PRNGs requires 35 elementary operations (17 XOR, 13 rotations, and 5 arithmetic operations) for generating a 32 bits number and thus $35 \frac{2n}{32}$ to get a number on $2n$ bits. Furthermore, since the strategy length is $2n$, the computation of S requires at most $k' + l' + 2n \times 35 \frac{2n}{32}$, which is less than $k' + l' + 5n^2$.

At least, since each iteration modifies only one bit, iterations require $2n$ elementary operations.

Finally, we have at most $2k' + 2l' + 5n^2 + 2n$ elementary operations to provide a hash value of size n . This bound is in $\mathcal{O}(l + n^2)$. ■

V. EXPERIMENTAL EVALUATIONS

Let us now give some examples of hash values before statistically studying the quality of hash outputs.

A. Examples of Hash Values

Let us consider the proposed hash function with $n = 256$. We consider the key to be equal to “my key”. To illustrate the confusion and diffusion properties [20], we use this function to generate hash values in the following cases:

Case 1. The original text message is the poem *Ulalume* (E.A.Poe), which is made of 104 lines, 667 words, and 3,754 characters.

Case 2. We change *serious* by *nervous* in the verse “Our talk had been serious and sober”

Case 3. We replace the last point ‘.’ with a coma ‘,’.

Case 4. In “The skies they were ashen and sober”, skies becomes Skies.

Case 5. The new original text is the binary value of Figure 1.



Figure 1: The original plain-image.

Case 6. We add 1 to the gray value of the pixel located in position (123,27).

Case 7. We subtract 1 to the gray value of the pixel located in position (23,127).

The corresponding hash values in hexadecimal format are:

Case 1. 0B4730459FBB5E54A18A9CCD676C8396
365B0104407D98C866FDAA51A07F0E45,

Case 2. 752E28088150B98166D870BC24177342
23A59463D44B83E9808383B30F8B8409,

Case 3. C10EED0A9D44856847F533E5647D0CCD
2C58A08643E4D3E5D8FEA0DA0E856760,

Case 4. 52BF23429EC3AD16A0C9DE03DF51C420
4466285448D6D73DDFB42E7A839BEE80,

Case 5. 5C639A55E2B26861EB9D8EADDF92F935
5B6214ADC01197510586745D47C888B8,

Case 6. E48989D48209143BAE306AC0563FFE31
EAB02E5E557B49E3442A840996BECFC1,

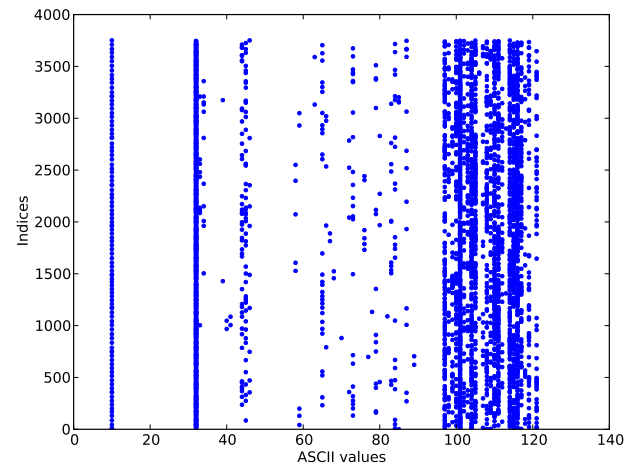
Case 7. EC850438A2D8EA95E691C746D487A755
12BEE63F4DDB4466C11CD859671DFBEB,

These simulation results are coherent with the topological properties of sensitive dependence to the initial condition, expansivity, and Lyapunov exponent: any alteration in the message causes a substantial difference in the final hash value.

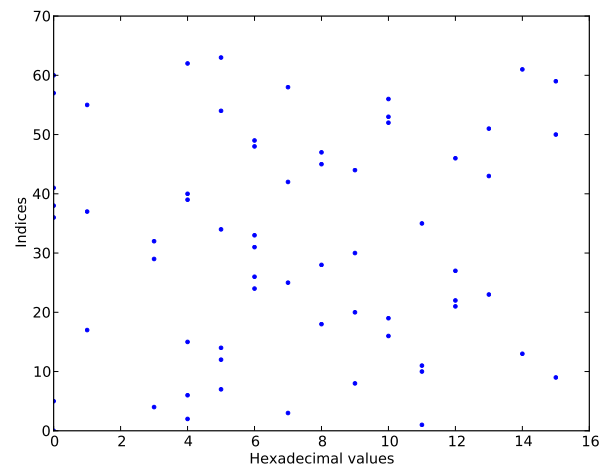
B. Statistical Evaluation of the Algorithm

We focus now on statistical studies of diffusion and confusion properties. Let us recall that confusion refers to the desire to make the relationship between the key and the digest as complex and involved as possible, whereas diffusion means that the redundancy in the statistics of the plain-text must be "dissipated" in the statistics of the cipher-text. Indeed, the avalanche criterion is a modern form of the diffusion, as this term means that the output bits should depend on the input bits in a very complex way.

1) *Uniform repartition for hash values:* To show the diffusion and confusion properties verified by our scheme, we first give an illustration of the difference of characters repartition between a plain-text and its hash value, when the original message is again the Ulalume poem. In Figure 2a, (resp. in Figure 2b) the X-axis represents ASCII numbers (resp. hexadecimal numbers) whereas the Y-axis gives for each X-value its position in the original text (resp. in the digest). For instance, in Figure 2b, the point (1,17) means that the character 1 is present in the digest at position 17 (see Case 1, Section. V-A). We can see that ASCII codes are localized within a small area (e.g., the ASCII "space" code and the



(a) Original text



(b) Digest

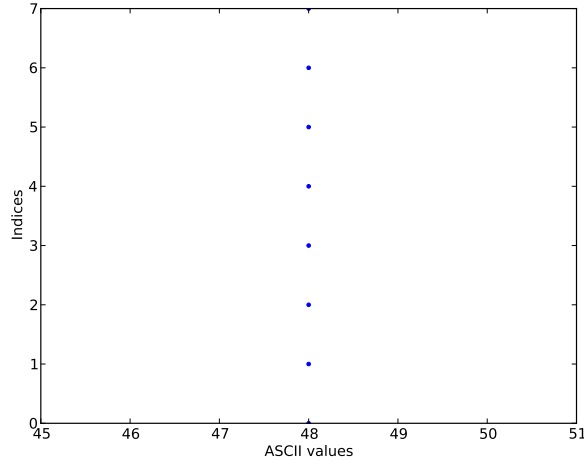
Figure 2: Values repartition of Ulalume poem

lowercase characters), whereas in Figure 2b the hexadecimal numbers of the hash value are uniformly distributed.

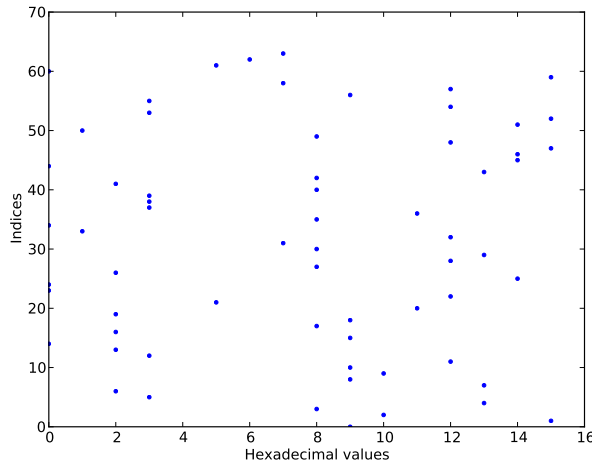
A similar experiment has been realized with a message having the same size, but which is only constituted by the character "0". The contrasts between the plain-text message and its digest are respectively presented in Figures 3a and 3b. Even under this very extreme condition, the distribution of the digest still remains uniform. To conclude, these simulations tend to indicate that no information concerning the original message can be found into its hash value, as it is recommended by the Shannon's diffusion and confusion requirements.

2) *Behavior through small random changes:* We now consider the following experiment. A first message of 1000 bits is randomly generated, and its hash value of size $n = 256$ bits is computed. Then one bit is randomly toggled into this message and the digest of the new message is obtained. These two hash values are compared by using the hamming distance, to compute the number B_i of changed bits. This test is reproduced $t = 10,000$ times. The corresponding distribution of B_i is presented in Figure 4.

As desired, Figure 4 shows that the distribution is centered around 128, which reinforces the confidence put into the good



(a) Original text



(b) Digest

Figure 3: Values repartition of the "00000000" message

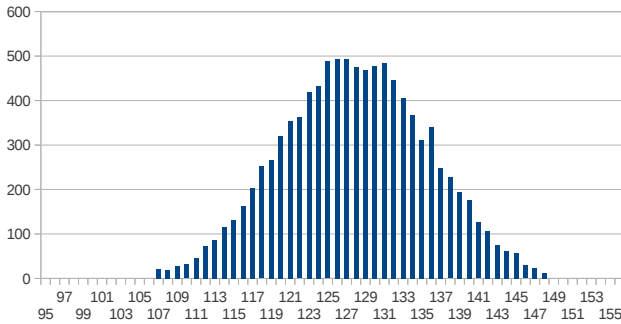


Figure 4: Histogram

capabilities of diffusion and confusion of the proposed hash algorithm. To analyse these results, the following common statistics are used.

- Mean changed bit number

$$\bar{B} = \frac{1}{t} \sum_{i=1}^t B_i.$$

	B_{min}	B_{max}	B	$P(\%)$	ΔB	$\Delta P(\%)$
$n = 256$	87	167	127.95	49.98	8.00	3.13
$n = 512$	213	306	255.82	49.97	11.29	2.21
$n = 1024$	446	571	511.54	49.96	15.97	1.56

Table I: Statistical performances of the proposed hash function

- Mean changed probability

$$P = \frac{\bar{B}}{n}.$$

$$\Delta B = \sqrt{\frac{1}{t} \sum_{i=1}^t (B_i - \bar{B})^2}.$$

$$\Delta P = \sqrt{\frac{1}{t} \sum_{i=1}^t (\frac{B_i}{n} - P)^2}.$$

The obtained statistics are listed in Table I where n belongs to $\{256; 512; 1,024\}$. In that study, starting from a message of length 1,000 and its digest, all the messages that have one bit of difference are further generated and the digest of the new message is obtained. Obviously, both the mean changed bit number \bar{B} and the mean changed probability P are close to the ideal values ($\frac{n}{2}$ bits and 50%, respectively), which illustrates the diffusion and confusion capability of our algorithm. Lastly, as ΔB and ΔP are very small, these capabilities are very stable.

C. Strict Avalanche Criterion Evaluation

This section focuses on checking whether the developed hash function verifies the strict avalanche criterion, as given in Definition 2. Quoting remarks of [15], "Unless n is small, it would be an immense task to follow this procedure for all possible vector pairs x and \bar{x}^i ". The authors propose thus the alternative method of computing a dependence matrix J of size $m \times n$ between the j -th, $1 \leq j \leq m$, element of the digest and i -th, $1 \leq i \leq n$, element of the original message. A simulation consists in first randomly choosing the size n of the message to hash (100 values in $\llbracket 1, 1000 \rrbracket$ for us). Next, a set of large size r ($r = 1,000$ in our case) of messages x is randomly computed. For each of them, the set $\{\bar{x}^1, \dots, \bar{x}^n\}$ is formed such that x and \bar{x}^i only differ in bit i . The set of m -bit vectors

$$\{f(x) \oplus f(\bar{x}^1), \dots, f(x) \oplus f(\bar{x}^n)\}$$

is thus computed where f is the hash function applied on vector of bits. The value of bit i (either a 1 or a 0) in $(f(x) \oplus f(\bar{x}^i))_j$ is added to J_{ij} . Finally each element of J is divided by r . If every J_{ij} are close to one half, the strict avalanche criterion is established. For all these experiments, the average value of J_{ij} is 0.5002, the minimal value is 0.418, the maximal value is 0.585, and the standard deviation is 0.016.

VI. CONCLUSION

In this research work, the hash function proposed in the Third International Conference on Evolving Internet, INTERNET11 (June 19-24, 2011, Luxembourg) [1] has been completely rethought. The second-preimage resistance has been proven, leading to better experimental results for the proposed hash function. Moreover, we have shown that this function has

a complexity that can be expressed as a polynomial function of the message length and of the digest size. Finally, we have statistically established that our function verifies the SAC.

If we now consider our approach as an asynchronous iterations post-treatment of an existing hash function. The security of this hash function is reinforced by the unpredictability of the behavior of the proposed post-treatment. Thus, the resulting hash function, a combination between an existing hash function and asynchronous iterations, satisfies important properties of topological chaos such as sensitivity to initial conditions, uniform repartition (as a result of the transitivity), unpredictability, and expansivity. Moreover, its Lyapunov exponent can be as great as needed. The results expected in our study have been experimentally checked. The choices made in this first study are simple: initial conditions designed by using the same ingredients as in the SHA-1, negation function for the iteration function, *etc.* But these simple choices have led to desired results, justifying that such a post-treatment can possibly improve the security of the inputted hash function. And, thus, such an approach should be investigated more largely.

This is why, in future work, we will test other choices of iteration functions and strategies. We will try to characterize topologically the diffusion and confusion capabilities. Other properties induced by topological chaos will be explored and their interest for the realization of hash functions will be deepened. Furthermore, other security properties of resistance and pseudo-randomness will be proven. We will thus compare the results of this post-treatment on several hash functions, among other things with the SHA-3 finalists [21].

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for their suggestions to improve the quality of the paper.

REFERENCES

- [1] J. M. Bahi, J.-F. Couchot, and C. Guyeux, "Performance analysis of a keyed hash function based on discrete and chaotic proven iterations," in *INTERNET 2011, the 3-rd International Conference on Evolving Internet*, Luxembourg, Luxembourg, Jun. 2011, pp. 52–57, best paper award.
- [2] J. M. Bahi and C. Guyeux, "Hash functions using chaotic iterations," *Journal of Algorithms & Computational Technology*, vol. 4, no. 2, pp. 167–181, 2010.
- [3] X.-M. Wang, J.-S. Zhang, and W.-F. Zhang, "One-way hash function construction based on the extended chaotic maps switch," *Acta Physica Sinica*, vol. 52, no. 11, pp. 2737–2742, 2003.
- [4] D. Xiao, X. Liao, and Y. Wang, "Improving the security of a parallel keyed hash function based on chaotic maps," *Physics Letters A*, vol. 373, no. 47, pp. 4346–4353, 2009.
- [5] —, "Parallel keyed hash function construction based on chaotic neural network," *Neurocomputing*, vol. 72, no. 10–12, pp. 2288–2296, 2009, lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Design (ISDA 2007).
- [6] D. Xiao, F. Y. Shih, and X. Liao, "A chaos-based hash function with both modification detection and localization capabilities," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2254–2261, 2010.
- [7] Wikipedia, "List of chaotic maps," Retrieved June, 2012 from http://en.wikipedia.org/wiki/List_of_chaotic_maps, 2012.
- [8] C. song Zhou and T. lun Chen, "Extracting information masked by chaos and contaminated with noise: Some considerations on the security of communication approaches using chaos," *Physics Letters A*, vol. 234, no. 6, pp. 429–435, 1997.
- [9] W. Guo, X. Wang, D. He, and Y. Cao, "Cryptanalysis on a parallel keyed hash function based on chaotic maps," *Physics Letters A*, vol. 373, no. 36, pp. 3201–3206, 2009.
- [10] C. Guyeux, "Le désordre des itérations chaotiques et leur utilité en sécurité informatique," Ph.D. dissertation, Université de Franche-Comté, 2010.
- [11] J. M. Bahi, J.-F. Couchot, C. Guyeux, and Q. Wang, "Class of trustworthy pseudo random number generators," in *INTERNET 2011, the 3-rd International Conference on Evolving Internet*, Luxembourg, Luxembourg, Jun. 2011, pp. 72–77.
- [12] J. M. Bahi, J.-F. Couchot, and C. Guyeux, "Steganography: A class of secure and robust algorithms," *The Computer Journal*, vol. 55, no. 6, pp. 653–666, 2012.
- [13] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, "Keyed hash functions," in *Cryptography: Policy and Algorithms*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1996, vol. 1029, pp. 201–214.
- [14] D. Coppersmith, "Another birthday attack," in *CRYPTO*, ser. Lecture Notes in Computer Science, H. C. Williams, Ed., vol. 218. Springer, 1985, pp. 14–17.
- [15] A. F. Webster and S. E. Tavares, "On the design of s-boxes," in *Advances in Cryptology*, ser. CRYPTO '85. London, UK, UK: Springer-Verlag, 1986, pp. 523–534.
- [16] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems, 2nd Edition*. Boulder, CO: Westview Press, March 2003.
- [17] C. Knudsen, "Chaos without nonperiodicity," *The American Mathematical Monthly*, vol. 101, pp. 563–565, 1994.
- [18] J. M. Bahi, R. Couturier, C. Guyeux, and P.-C. Héam, "Efficient and cryptographically secure generation of chaotic pseudorandom numbers on gpu," *CoRR*, vol. abs/1112.5239, 2011.
- [19] C. Guyeux, N. Friot, and J. Bahi, "Chaotic iterations versus spread-spectrum: chaos and stego security," in *Sixth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2010)*, Darmstadt, Germany. Washington, DC: IEEE Computer Society, Oct. 2010, pp. 208–211.
- [20] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, 1949.
- [21] M. S. Turan, R. Perlner, L. E. Bassham, W. Burr, D. Chang, S. jen Chang, M. J. Dworkin, J. M. Kelsey, S. Paul, and R. Peralta, "Status report on the second round of the sha-3 cryptographic hash algorithm competition," National Institute of Standards and Technology, Tech. Rep. NIST Interagency Report 7764, february 2011, retrieved June, 2012 from http://csrc.nist.gov/groups/ST/hash/sha-3/Round2/documents/Round2_Report_NISTIR_7764.pdf.

A Reference Model for Improving an Inter-Organizational IT Management Tool

Mark Yampolskiy
Vanderbilt University
1025 16th Ave S, Nashville
TN 37212, USA
Email: myy@isis.vanderbilt.edu

Silvia Knittl
msg systems ag
Robert-Bürkle-Straße 1
85737 Ismaning/München, Germany
Email: silvia.knittl@msg-systems.com

Feng Liu
Leibniz Supercomputing Centre (LRZ)
Boltzmannstraße 1
85748 Garching, Germany
Email: liu@lrz.de

Abstract—In recent years, the number of collaborations between IT service providers beyond national borders has been increasing tremendously. From the organizational point of view, all participants of such collaboration are independent domains. The participation of organizations in such collaborations is motivated by different factors. These factors include, but are not limited to, specialization on different technical domains, cost-optimization through usage of shared infrastructure and services, and regulation policies imposed by national laws. The IT service provisioning is always fostered by IT service management. In order to enable the collaborative service provisioning, the IT service management should be able to operate across organizational borders, including coordination of intra-domain activities. The capability to provide a holistic view of service infrastructures is one of the key factors that contribute to the success of an inter-domain collaboration. As a consequence, relevant information and events should be shared among the involved domains. In this paper, we present the use case driven tool I-SHARe as well as the theoretical framework inter-organizational (io) CMDB, both focused on the information sharing among collaborating organizations. The results of the I-SHARe pilot deployment in the pan-European collaboration Géant proves the necessity of such tools. Meanwhile, the evaluation of I-SHARe against requirements elaborated in the ioCMDB reference model identifies the tool's significant optimization and improvement potential.

Keywords - IT Management, IT Service Provider Collaborations, CMDB, e-Infrastructures

I. INTRODUCTION

In our previously published work [1] we discussed the necessity of developing an inter-organizational information exchange system using the Géant project as an example. Géant is a pan-European collaboration of over 30 national grade network providers, also known as National Research and Educational Networks (NRENs). Under this collaboration, NRENs provide the network infrastructures for international research projects such as the world's largest particle physics project, the Large Hadron Collider (LHC), located at CERN, or Grid collaborations such as Enabling Grids for E-science (EGEE). The Géant collaboration offers networking services ranging from conventional IP connectivity with the best-effort connection quality up to dedicated optical End-to-End Links (E2E Links), which are provisioned to deal with data deluge that needs to be transported over networks.

Establishment of dedicated E2E Links per user requests often involves activities including planning, procurement, in-

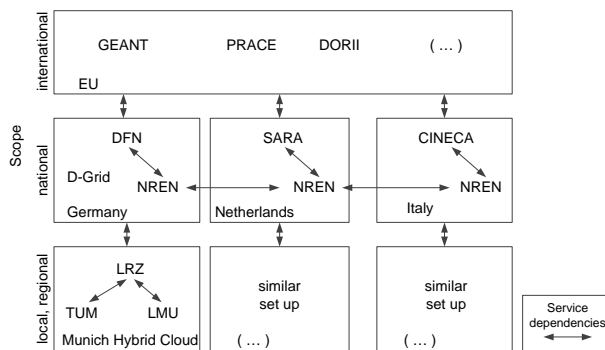


Fig. 1. Organizational scope of e-Infrastructure projects

stallation and configuration of the required network devices. In order to avoid deadlocks in such a rather complex process and guarantee the compatibility of involved devices, a tool support for the information sharing becomes a crucial influencing factor for a successful provisioning process. I-SHARe (Information Sharing across Heterogeneous Administrative Regions) is specifically designed and implemented to fulfill this purpose.

As a successor to our previous work, in this paper we not only present the in-depth knowledge about I-SHARe, we also intend to provide a retrospective view and empirical experiences gained after its pilot operation. Most importantly, we systematically identify improvement potentials of the current version of I-SHARe for the future development. These could be regarded as improvements that are complementary to the user requests we received. The applied analytical methodology is based on the well-established design science [2]. Aligned with this method we introduce our problem field based on the actual observations of the tool aspects in the Géant case, for which I-SHARe is designed and implemented for a better support of IT service management. Although our discussion is conducted based on the Géant scenario, our findings are not restricted to this special use case. We see the possibility to reuse our results in other tools aiming to support inter-organizational IT service management (ioITSM) in collaborations. Potential application examples for further e-Infrastructure related projects are illustrated in Figure 1.

As shown in the figure, the institutional scope of the listed projects is divided into regional, national and international

levels. The arrows indicate dependencies both within (e.g., NREN to NREN) and between the layers (e.g., Géant to DFN to LRZ). The Leibniz Supercomputing Centre (LRZ) for example is involved in e-Infrastructure projects on every level, like the Munich Hybrid Cloud environment on the local level with participation of the Munich universities (LMU, TUM), Grid projects (D-Grid) on the national or the Géant and PRACE project on the international level (see also [3][4]). In other countries similar institutional setups can be found. These listed projects have been previously strictly focused on operational aspects. With ever-increasing number of user base and scope, ioITSM issues become more compelling. Thus, a professionally operated IT service management with appropriate tool support is needed more than ever.

The rest of this paper is organized as follows: after giving a rather detailed discussion on a case study based on the Géant project in Section II, we outline our tool concept to assist ioITSM in Section III. This concept is mainly based on the results described in [5] and can be used as a reference model to either build a new tool from scratch or to evaluate existing tools according to their possible usage as an ioCMDB. We then discuss the latter aspect in Section IV. Our discussion mainly concentrates on a systematic analysis of an inter-organizational management approach developed in the Géant project. A brief survey on the related work is provided in Section V. We conclude this paper with a perspective for the future work in Section VI.

II. CASE STUDY: I-SHARE IN GÉANT

In this section we first outline the challenges by establishing and operating Géant network service *E2E Links*. Then we present an information sharing tool called *I-SHARE*, which is developed in the Géant project to support and coordinate manual handling processes of multi-domain E2E Links. We conclude this section with a discussion of experiences gained during the pilot operation with I-SHARE and our further development plan. The discussion provides the basis for the understanding of the challenges faced by the inter-organizational information sharing tools.

A. Géant Service E2E Links

The purpose of Géant is to interconnect NRENs and therefore to foster international research projects, in which participating organizations are connected through different NRENs. The portfolio of Géant includes various services among others conventional IP connections. However, such services cannot always fulfill all the challenging requirements of modern research collaborations. One of the most prominent examples of such a challenge is the Large Hadron Collider (LHC) project, which produces over 15 petabytes of raw experimental data per year [6] and related Worldwide LHC Computing Grid (WLCG) [7] built with the purpose of data processing and analyzing for the LHC experiment.

For a dependable and robust operation, such projects often rely on network connections with rigorous quality assurance. Realizing high-quality high-bandwidth connections in general

purpose IP networks is a challenging task, many reasons can deteriorate network quality, for example communication flows can interfere with each other and lead to an inferior connection quality. In order to cope with user's challenging demands, a novel End-to-End (E2E) Link service has been introduced in Géant. E2E Links are dedicated optical point-to-point connections realized at ISO/OSI layers 1 and 2, with connection segments provided by one or more NRENs [8]. An E2E Link across multiple domains differentiates from its single domain counterpart in its quality requirements, variety on the participating networking technologies and geographical dimensions. E2E Links are multi-domain backbone connections, in which – in opposite to classical backbones – multiple network providers are involved and heterogeneous network technologies can be used. The E2E Link structure is presented in Figure 2 in principle.

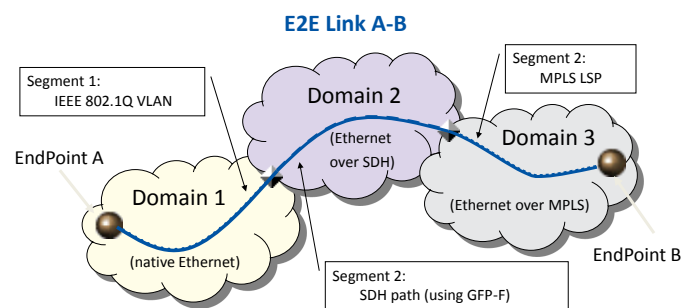


Fig. 2. Typical E2E Link Structure [9]

A main feature of a multi-domain E2E Link service is that a new connection can be requested and ordered independently from the actual availability status of required infrastructures. If a new infrastructure is needed to sustain the customer's request, it can be procured, installed, and configured according to the corresponding requirements of the new E2E Link. Consequently, all route planning procedures can only be done manually and require intensive interactions and task trackings between the involved NRENs.

Empirical experiences gained during the first years of the E2E Link service has revealed that exchanging and maintaining connectivity information determine the time efficiency of planning and installation of a particular E2E Link across domains. Information exchange via e-mail and planning via Excel sheets has proven to be error prone with a high probability of information losses or missing events, e.g., the delivery of the procured infrastructure by neighboring NRENs. This causes a high fluctuation on the time needed to plan and install new links. In order to improve this situation, a tool supporting the information exchange among participating NRENs was introduced for the E2E Links service.

B. Sharing Information with I-SHARE

The design and development of the *I-SHARE* tool has been performed by an international team of researchers working for different NRENs. I-SHARE covers information exchange for

a complete life cycle of an E2E Link service instance, from its planning up to the decommissioning of the link.

Handling of single-domain and multi-domain information is clearly distinguished in I-SHARE at system architecture level (see Figure 3). Information such as operational groups and group members, their responsibility areas and contact data are handled in the *domain part* or in a NREN's domestic management tool. In both cases the single-domain information is propagated to I-SHARE via the *I-SHARE Domain Interface*. *I-SHARE Central Server* stores the copy of the provided information, so that it can be incorporated in the supported processes.

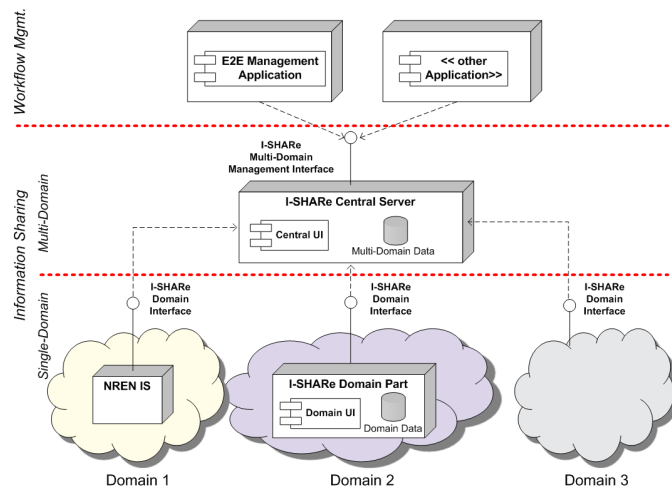



Fig. 3. System architecture of *I-SHARE* [10]

Multi-domain information like the route of an E2E Link through NRENs, interfaces of the adjacent connection parts provided by neighboring NRENs, and states of various operations are stored directly in the *I-SHARE Central Server*. This information can be accessed and edited via a web-based GUI.

All information stored in *I-SHARE Central Server* can be accessed from other applications through the *I-SHARE Multi-Domain Management Interface*. This interface is designated to facilitate the integration of I-SHARE with other tools, e.g., with other workflow management systems or analysis tools, so that relevant information regarding network connectivity can be reused and shared among different tools.

I-SHARE is designed to support the multi-domain manual management processes for a whole life cycle of E2E Links. According to specifications of the E2E Links service, one has to distinguish between four phases: (i) *Ordering* of a new E2E Link, (ii) *Setting up* of the ordered E2E Link, (iii) *Operation* of E2E Links in-service, and (iv) *Decommissioning* of no longer needed E2E Links. E2E Links in different phases can be accessed through different views (see top-level tabs in the GUI in Figure 4) of the *I-SHARE Central Server*.

Views on different phases of an E2E Link must be distinguished as the tasks in the corresponding phases require different knowledge, skills and competences, which are generally provided by different teams. Furthermore, the information


1.0

E2E Link Requests

[Logout](#)

[All](#)
[Ordered](#)
[Set Up](#)
[Operational](#)
[Decommissioned](#)
[Action-Log](#)

Request Information

Action Status

ID	Project	End Site A	End Site B	Ordered on	Ordering Coordinator	Route Finding	UNI Negotiation	NNI Negotiation	Offer To The End Site	Acceptance	Set Up Coordinator
<< Previous 1/1 Next >>											
3	LHCOPN	CERN	CNAF								
4	DEISA	CINECA	FRA								
5	LHCOPN	CERN	CNAF	2010-11-01		✓	✓	✓	✓		
6	LHCOPN	PIC	CERN	2011-01-31							
7	DEISA	BSC	FRA	2011-01-31							

Add

Fig. 4. I-SHARE's list of ordered links [11]

E2E Link ID: CERN-CNAF-LHCOPN-001

Set up start: 2010-10-25

Estimated delivery: 2010-10-25

Global

End Sites	NRENs	CERN	CERN <=> GÉANT	gen-mil_LHC... (GÉANT)	GÉANT <=> GARR
✓	✓	✓	?	✓	✓

Fig. 5. Install and configure the network infrastructure [11]

needed in various phases are logically arranged in a sequential manner and overlap only partially. Therefore every view contains the list of E2E Links in the particular phase as well as set of status check boxes specific for the particular phase.

In the detailed view of a single E2E Link, only actions that are specific for the particular phase can be performed. For instance, during the ordering phase the E2E Link route through NRENs as well as interconnection points between NRENs (so called *Demarcation Points*, DPs) can be specified. During the installation phase this information can be refined with further details. For example the progress on – or difficulties of –

installation can be specified for every single interface (see Figure 5).

One of the most important advantages of I-SHARe is the possibility to coordinate efforts between NRENs. As all NRENs are independent organizations and often use hardware from different vendors and/or prefer different network technology, the compatibility of interconnected interfaces becomes one of the most critical factors.

Even if I-SHARe is designed as an information exchange rather than a workflow tool, however, some operation logic restrictions are already integrated. For instance, the button for declaring an E2E Link operational becomes enabled only after checkboxes of all involved interfaces are set to the state indicating that the setup is completed.

C. I-SHARe: Empirical Experiences and Future Plans

A detailed description of I-SHARe version 1.0 can be found in [1][10]. After the initial release in 2010, a user survey has been conducted among selected NRENs during the pilot phase of operation. Feedbacks obtained from participating NRENs show a high acceptance of I-SHARe for its functionalities and usability. However, the survey also reveals several deficiencies that the users wish to be improved in the next version [12].

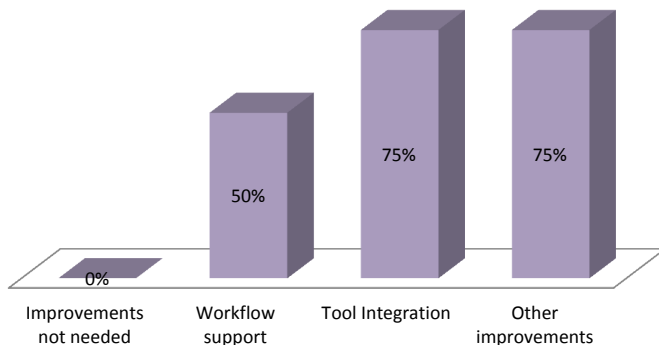


Fig. 6. Survey result on user requested improvements [12]

Among the requested I-SHARe improvements workflow support and tool integration are weighted as the most important feature (see Figure 6). As I-SHARe is a supporting tool, it should be flexible enough to sustain established workflows in the project. It also implies that the tool should be adaptive to changing workflows, which is a common case as shown by empirical experiences. For instance, during the time elapsed from the requirements analysis until the deployment of the tool in its pilot phase, the role model used in manual workflows has been changed. This change has introduced the necessity to change the role model implemented in I-SHARe.

Tool integration is a further critical issue that requires careful consideration. As I-SHARe supports manual processes, it should prevent users from the daunting tasks of re-entering same data in different tools. A technical solution for this problem is an extended communication interface. This interface should allow external tools to communicate with I-SHARe

V2.0 in order to access existing or publish new data. For instance, it is planned that the E2E Link monitoring software *E2Emon* [13] will access the information about physical links from I-SHARe. In contrast, in the future it should be also possible to propagate the information about planned E2E Link segment maintenance from NREN-internal tools to the I-SHARe so that this information can be seen and considered in the multi-domain planning.

The pilot survey has also disclosed the necessity for several new features that were not identified during the initial requirements analysis process. For instance, in I-SHARe V2.0 a further communication capability should be implemented, so that stakeholders of a particular links can be automatically informed about the current status via e-mail notification in cases of network anomalies. This feature allows I-SHARe to provide assistance to the manual management processes even if the network operators are not necessarily logged into the tool.

III. ioCMDB REFERENCE MODEL

Motivated by the aforementioned use case, there is a compelling need for a tool support of ioITSM. I-SHARe is designed and developed to address such needs. In order to systematically identify and formally derive improvement potentials of I-SHARe, in this section we apply a reference model of ioCMDB based on the concept discussed in [5]. This concept is fully in alignment to de-facto standards of ITSM, such as IT Infrastructure Library (ITIL, see: [14] and Section V). As suggested by ITIL, a Configuration Management Database (CMDB) acts as an information nexus to provide relevant management information to all disciplines of ITSM. The entities stored in a CMDB are Configuration Items (CI), which represent information about software, hardware, services, related documents and their corresponding interrelations. A readily available information repository as such will greatly improve the efficiency of IT management. Incident management, for example, uses specifications of CI priorities for the controlling of waiting queues, Change management is able to anticipate the impact of planned changes on potentially affected CI by having an overview of the CI relationships, and Availability management is able to identify the occurrence of single points of failures [15].

Whereas the conventional CMDB concept is well-studied and mature for IT services in a single organization, a comparative approach need to be elaborated and advised to take inter-organizational aspect into consideration. Our efforts toward this kind of approach is called inter-organizational CMDB (ioCMDB).

The applied research method is based on the principles of design science [2], which is a formal approach to build artifacts. Our claim is to create a reference model of an ioCMDB. As a reference model possesses both descriptive as well as prescriptive characteristics, the requirements identified within the design process in [5] will be used as a basis for the evaluation of I-SHARe. This evaluation will show whether I-SHARe is an appropriate tool to serve the purpose of

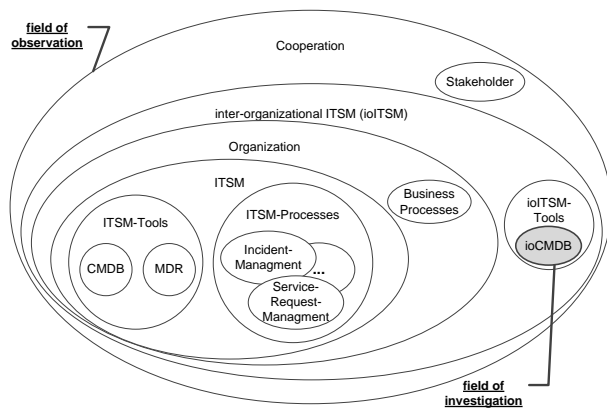


Fig. 7. Field of investigation and observation

information-sharing across the organizational boundaries. The more elaborated discussion is presented in Section IV. Note that, besides its critical role in building a new system complete from scratch, a reference model also prevails in evaluating an existing framework [16]. The initial step in our methodology is to define a frame of reference, in which a relationship between ioITSM and ITSM is intended to be established. Figure 7 illustrates a view on the fields of investigation and observation. As shown there, our investigation comprises all elements relevant for the designing goal of ioCMDB. The field of observation contains all related topics, which influence our investigation. Those include, e.g., relationships to other tools, ITSM processes, business processes, organizational set up, etc. The fundamental need for an ioCMDB is not limited to the problems as we demonstrated in the Géant case, we also observe the same problem area in similar structured IT service provider collaborations [3][17]. For an ioCMDB, a comprehensive list of requirements has been elaborated in [5]. All these requirements are structured according to the management architecture domains defined in [18] into information, organization, communication and functional model related requirements. In this way, all derived requirements are structured into manageable building blocks. A crucial input of this analysis is a set of ioITSM processes observed from various case studies and investigations, e.g., inter-organizational Fault or Change management [19][20].

In Section IV we demonstrate the usage of our ioCMDB reference model as a basis for evaluating the I-SHARE tool.

IV. MAPPING BETWEEN ioCMDB REQUIREMENTS AND I-SHARE

In our current work, we apply the elaborated ioCMDB requirements on the actual I-SHARE implementation. We summarize the results of the evaluation in a table form. For conciseness, we show here merely the evaluation results for a selected subset of the requirements. The results of the evaluation are presented on the right side of the column named "I-SHARE." Every evaluation entry consists of two ratings: on the left side of an arrow, the evaluation of the current implementation is presented; on the right side, the rating of I-

SHARe after implementing planned changes is presented (see Section II-C for the description of planned changes). Thus, the rating scheme in the last column of the tables can be read as

$$[actual\ rating] \rightarrow [improved\ rating].$$

This comparison will also show, that not all potential aspects of optimization have been identified within the internal Géant evaluation itself. Therefore, the remaining gaps identified in this paper can then be used by the Géant project management as a foundation for the definition of a new I-SHARE development road map.

A. Information model related requirements

An essential prerequisite of IT service management is to have an overview of the entities being managed as well as their dependencies. Such information is necessary for assessing, e.g., the effects of future changes. These aspects are covered within an information model (IM). The IM enables a logical view on the IT infrastructure, even if this infrastructure is stretched over multiple domains. In Table I the results of the evaluation with respect to an IM of an ioCMDB are listed. A description of the foundation of our information modeling concept, which is based on established standards in the Business-to-Business (B2B) environment is outlined in [17].

Label	Requirement	I-SHARE
IM-1	Ability to model CIs	0 → +
IM-2	Ability to model Interrelationships	0 → +
IM-3	Ability to model dynamic information	0 → +
IM-4	Ability to model meta data	+ → +
IM-5	Ability of unique identification	0 → 0
++ : fully covered, +: partly covered, 0: not covered at all -: irrelevant for this Scenario		

TABLE I
REQUIREMENTS CONCERNING INFORMATION MODEL

1) *Evaluation of information model requirements:* An important requirement an ioCMDB has to meet is its capability to model different classes of CIs and different types of inter-relationships between these CIs. The IM for planning E2E network connection with quality assurance is shown in [21], in which specific CIs and their inter-relationships are described, e.g., `Domain`, `LinkProperties` or `CompoundLink`. In the current I-SHARE implementation such kind of entity information is predefined and hard coded. Thus, a corresponding Configuration Manager is not capable of defining new or alter existent CI types. Consequently if new types are requested the programmers have to extend them accordingly, build a new release of I-SHARE and make a roll out of this release in all participating organizations.

Another requirement concerning IM is the possibility to apply meta data, i.e., the data about the data. These can be distinguished according to [22] in the following functional areas:

- administrative, like data about the source information; in our case it would be the service access point of the

connected domain specific CMDDB or rather any other Management Data Repository (MDR),

- descriptive, e.g., annotations by users,
- preservation, like data refreshing cycles,
- technical, as for example data related to how a system functions like it's software documentation,
- use, metadata related to the level and type of use of information resources, like log files.

In the actual I-SHARe release, meta data used for further description of extrinsic or intrinsic features of CIs like owner or status lifecycle are hard coded and thus, evaluated to 'partly applicable' in Table I.

2) *Identified improvements in information modeling*: The need for being flexible in defining dedicated CI classes has been identified as an action item to be implemented in the next release of I-SHARe. However, in contrast to our ioCMDDB requirement for having an explicit role model, i.e., for the role of an inter-organizational Configuration Manager, any person is allowed to create CI types. As it is assumed presently that the tool is operated in a environment of dedicated services, currently such CI types are only representing the type interface. Thus, the evaluation remains on 'partly applicable'. According to our concept, a more dynamic modeling means must be implemented in I-SHARe to support various cross-organizational scenarios and services. In the actual improvement plan, the need for having meta data in place to improve the I-SHARe's operational capabilities is not recognized. This is the reason, for which the rating will remain the same for the next planned release cycle.

B. Organizational model related requirements

While IM addresses the modeling aspects of CIs and their relations, the focus of an organizational model (OM) lays on the governance structure. This is especially important for our intended use cases of cross-organizational provider collaborations. Thus, OM aspects have to be reflected in an ioCMDDB. The requirements for a mapping of this structure as well as the adequate mapping of the user's roles are met to a high degree by the actual implementation of I-SHARe as can be seen in Table II.

Label	Requirement	I-SHARe
OM-1	Mapping of the governance structure	+ → +
OM-2	Mapping of roles/users	+ → +
++: fully covered, +: partly covered, 0: not covered at all -: irrelevant for this scenario		

TABLE II
REQUIREMENTS CONCERNING ORGANIZATIONAL MODEL

1) *Evaluation of organizational model requirements*: The governance structure within the actual I-SHARe release allows mapping users to groups and groups to roles. These roles are hard coded. Thus, this fact is rated as 'partly covered'. Due to organizational changes in such environments the corresponding governance structure might change as well. This aspect has not yet been addressed within the project.

2) *Identified improvements in organizational model*: The need for having roles in place for enabling a dedicated management of controlling issues has been identified also within the project itself. The actual release planning in the I-SHARe project is discussing the implementation of a transaction based role model. This will allow a transaction and role based logging, which again permits analyzing of log files. But here again, the role model itself will remain hard coded, and won't be changeable on a flexible base as suggested by our requirement catalogue.

C. Communication model related requirements

The mechanism, how management information are exchanged between the ioCMDDB and related local CMDDBs are subject of the communication model. In Table III the requirements concerning communication mechanism are shown.

1) *Evaluation of communication model requirements*: One important issue is the channels the exchanged messages of management information are taking. It is a requirement, that such channels can be specified according to the corresponding need. We will further describe the interaction characteristic in more detail in future work. At the moment, this requirement is not fulfilled completely by the actual I-SHARe release. This leads to the fact, that I-SHARe cannot support ioITSM processes efficiently. At present the requirement KM-2 for having web based communication means is fully covered by using e-mail and web services. Although, there is no way to define the communication characteristics as requested by requirement KM-4. Having a registry mechanism in place, where new or changed members or roles can be registered is of minor relevance for this scenario, since we are facing a stable organizational structure and its changes is the matter of long term planning.

Label	Requirement	I-SHARe
KM-1	registry	-
KM-2	web based communication	++
KM-3	define direction of communication	0 → ++
KM-4	define communication mechanism	0 → +
++: fully covered, +: partly covered, 0: not covered at all -: irrelevant for this scenario		

TABLE III
REQUIREMENTS CONCERNING COMMUNICATION MODEL

2) *Identified improvements in communication modeling*: According to the actual I-SHARe release planning, it is intended to send messages containing information about, e.g., state changes of inter-organizational relevant CI like the status of an E2E Link within the order process via e-mail to all participating members within Géant. The disadvantage of such a mechanism is a potential information flooding of all the recipients instead of informing only the affected responsible persons, resulting in rather ignoring such e-mails than reacting consequently.

Due to the requirements identified in [5] it should be possible to define both the direction of the communication and the mechanism, i.e., in case of state changes only the responsible

roles should be informed via appropriate mechanism, which can be for example alerts or warning e-mails. The need for defining the communications directions, e.g., information provided by a local partner, that is propagated to I-SHARE and from there to relevant recipients in other organizations has also been identified as an open issue in the project. Thus, the introduction of roles should enable to support actions of the I-SHARE Domain Part, where an interface allows for the subscription of dedicated information with either push or pull mechanism. The need for having such subscription mechanism in place has also been recognized and is actually under discussion.

D. Functional model related requirements

Table IV outlines functional requirements a tool has to meet, if it is used as an ioCMDB. To assist ioITSM via an appropriate tool, it has to cover the lifecycle issues both of organizational as well as information related entities. Thus, the manipulation of roles and CI, the support of state changes of the cooperating partners, the possibility of adding, changing or deleting CI are vital aspects. Further, the users of the tool can have an easier insight at the management information, if there is appropriate visualization means offered.

These functional requirements are evaluated below.

Label	Requirement	I-SHARE
FN-2-2	manipulation of roles	0 → 0
FN-2-3-0	manipulation of configuration items	
FN-2-3-1	maintenance of configuration items	0 → ++
FN-2-5	visual representation of content	++
FN-2-6-1	data maintenance: transformation rules	0 → ++
FN-2-7	data maintenance: interfaces	+ → ++
++: fully covered, +: partly covered, 0: not covered at all -: irrelevant for this scenario		

TABLE IV
REQUIREMENTS CONCERNING FUNCTIONAL MODEL

1) *Evaluation of functional model requirements:* The current rating of the I-SHARE tool with respect to the functional model necessities can be seen in the Table IV. Examples of the I-SHARE information visualization are illustrated above (see Figures 4 or 5). The maintenance of configuration items itself, like adding new CI and their interrelations are implemented as quite complex procedures at the moment. Maintenance processes for filling in ioCMDB contents automatically are important in complex environments, since manual changes are leading to higher error rates and are raising the possibility of having not synchronized data pools. To update the contents in I-SHARE, no automatic updating mechanism between other tools like CMDBs from the participating members exist. Almost all data have to be maintained by hand. A CI that has been associated with a dedicated type of relationship can't be changed, once it has been introduced. In case, there would be automatic updating procedures in place, such procedures need to have a transformation mechanism in place resulting from autonomously defined data schemes of existent tools at the members' site. Such transformation rules are not covered

at all by the actual I-SHARE implementation. One additional aspect is to identify the ownership of the data. This can be also part of visualization aspects [23]. Other Interfaces, as the ones for the e-mail based notification mechanisms, are currently not in place.

2) *Identified improvements in functional model:* The further release planning of the I-SHARE tool is addressing the need for having transformation rules in place. Then, an information model will be defined on a global level and all necessary transformations have to be implemented at the level of the member sites. This will change the future rating to 'fully covered'. The need for having better interfaces in place has been discovered. New releases will have interfaces, e.g., to monitoring tools and thus, the rating will be raised to 'fully covered'.

E. Nonfunctional requirements

Table V outlines an excerpt of non-functional requirements, i.e., all requirements concerning the operation and usage of the ioCMDB regarding the I-SHARE tool. A substantial need is to have access control and audit mechanisms in place to improve security issues, since the ioCMDB users are associated with the collaboration partners and acting in different roles. We have proposed an access control solution based on federated identity mechanism in [24]. Audit mechanism facilitates inspections resulting from compliance or regulatory requirements. By taking cultural considerations into account as for example different language versions or other dedicated cultural characteristics the user acceptance can be certainly improved.

Label	Requirement	I-SHARE
NF-4-1	security: access control	+ → +
NF-4-2	security: audit mechanism	+ → ++
NF-7	cultural requirements	0 → 0
++: fully applicable, +: partly applicable, 0: nonexistent		

TABLE V
NONFUNCTIONAL REQUIREMENTS

1) *Evaluation of nonfunctional requirements:* The actual state of access control in I-SHARE only allows authentication. Authorization means are missing and therefore, a rating to 'partly applicable' is justified. Since there is no authorization, audit mechanisms are used instead to be able to afterwards reconstruct performed transaction. Although Géant is an international project, no country specific characteristics, like different language or currency codes, have been introduced.

2) *Identified improvements in nonfunctional modeling:* In the near future access control mechanisms and cultural requirements are not planned for I-SHARE. Thus, the evaluation will remain on the same level. Cultural topics are considered of minor importance because all participants are expected to understand the English version of I-SHARE. The audit mechanisms have been undertaken an additional review process and will be improved both in structure as well as usability.

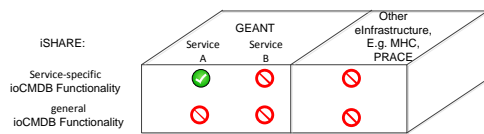


Fig. 8. As is portability of I-SHARE

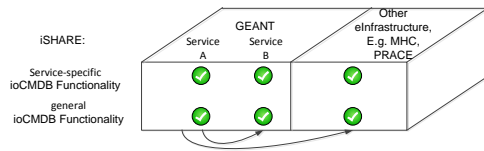


Fig. 9. To be portability of I-SHARE when adhering ioCMDB concept

F. Evaluation summary

In this section, we evaluated I-SHARE based on the requirements established for an ioCMDB. Further, we have considered the actual release planning of the I-SHARE. Despite its success in the pilot operation phase, there is still room for improvement observable for I-SHARE, as we showed and discussed in great details in this section. Such improvements will mainly affect scalability issues, as have been for example identified within the information model needs of having variable mechanisms of mapping new and changed types of CIs and relationships. The lack of having such mechanisms implemented leads to the fact that the actual I-SHARE is supporting the management of only a single type of service at the moment. Other ITSM related management processes have to be implemented separately.

Our evaluation shows that even if there have already been many desirable extensions identified in the Géant project, there is still potential for further improvement. Such improvement potentials need to be discovered and approached systematically. The actual usage spectrum of I-SHARE is shown in Figure 8. I-SHARE has been designed to support service and project specific purposes. Consequently, resulting in poor means of portability to manage other services as well and therefore, making it difficult to use it a) for other services respectively IT service management processes within the Géant project and b) for other similar structured projects, which have the same requirements. Indeed, there can be no synergy effects found to reuse this platform for ITSM related purposed within other e-Infrastructure projects as have been illustrated above.

If the concepts created in [5] will be used for further extension of the I-SHARE tool, a tremendously better portability can be established as can be seen in Figure 9. Consequently, this platform can be used to support cross-organizational IT Service management within other projects as well.

V. RELATED WORK

Having introduced the reference model for the ioCMDB above, in this section, we now review some key researches, including existing frameworks, methods and models, which are relevant to our work.

Well-established frameworks exist in the realm of ITSM, one of the prominent examples is the IT Infrastructure Library (ITIL) [14]. In ITIL the concept of the Configuration Management Database (ITIL version 2) or Configuration Management System (ITIL version 3) has been introduced as an information platform supporting all other ITSM disciplines. Nevertheless, although there are some marginal descriptions concerning outsourcing relationships, ITIL itself does not consider the cross-institutional case in terms of collaborations between IT service providers. Furthermore, even if a CMDB should possess additional interfaces for tools from external suppliers, no available proposal concerning the design of such interfaces is currently known to us.

Other configuration management researches verify necessities of a new concept for an ioCMDB. The suggested information models for building up an intra-organizational CMDB are however not suitable for direct applications in real-world usages. CMDB tools currently applied in large scale intra-organizational environments do have – in the most cases – a focus on the visualization of monitoring aspects. Issues regarding global writing access to distribute data sources in a federated environment have not yet been fully covered [25]. Network specific CMDB tools that are capable to integrate data from different sources [26] are not directly applicable to our use cases, due to its lacking of considerations on the inter-organizational information models and functional requirements as we have identified.

A concept of CMDB federation (CMDBf) is proposed in [27]. The main focus of the suggested approach is on the integration of tools of different vendors (Management Data Repositories, MDR) into a single CMDB. To serve that purpose, a set of web-services are specified to facilitate data communication within a single organization. Concerning our requirements on the communication model, the suggested approach provides an useful reference and can be extended as a viable building block for the inter-organizational scenario. The extension should concentrate on implementing the CMDBf interface for all connected tools. According to the analysis in [28] the CMDBf will become a standard for heterogeneous CMDB integration and federation by the year 2013. But as it has been stated in [29] this standard only covers the data exchange aspect without specifying data types and their corresponding information models.

According to [30] the CMDBf is based on having a central management CMDB in place, it does not cover use-cases in cross-domain environments, e.g., in Cloud computing. Due to the fact that an universal access to any CMDB cannot be presumed in real-world IT operations, the authors propose a cross-organizational CMDB concept, in which a domain can expose individual CI information as RESTful web services. These resources can then be referred to and read by other domains in the context of service management processes. Nevertheless, this concept concentrates solely on communication related issues. Aspects of inter-organizational information modeling and a concept of authentication or authorization have not been considered.

In [31][32] concepts of federated or distributed databases are introduced. However, the design goal of ioCMDB is to assist the inter-organizational ITSM (ioITSM) processes and workflows. Despite its crucial role in ioCMDB, a distributed or rather federated database cannot be directly applied as one might expect it. Even if they share some similar characteristics such as distributed locations, autonomy and heterogeneity, however the access at database level to every connected CMDB or MDR still cannot be always presumed. Contrary to their approaches, dedicated interfaces between ioCMDB and CMDB or MDR must be implemented in our proposed solution. Instead of manipulating the contents of ioCMDB at databases level, our approach suggests to perform such alternations through specified functional areas, as we outline in Section IV-D. In this way, an alignment between ioCMDB contents and specifications of information models can be ensured.

Finally, data warehouse technique [33] has been applied as a viable approach to support complex analytical processes. Even though some commonalities can be identified between data warehouse and our use cases, e.g., analysis in incident or change management, however, data warehouse is optimized for read-only access. Modifying data at the global level and propagating such modifications to all connected participating CMDBs, as required by our scenario, are not well-supported by data warehouse.

VI. CONCLUSION

Recent proliferation of large scale IT-collaborations at a global level require highly efficient processes to cope with intricate tasks of infrastructure management. Whereas best-practice recommendations such as ITIL provide guidelines for designing management processes within a single organizational unit, more elaborations and research are needed to investigate issues regarding the management across multiple organizations. They not only differ in their geographical locations, they are also administratively independent entities, which exaggerate the difficulty to design and deploy efficient management processes. In fact, the management of infrastructures across organizational boundaries has not yet been fully understood and thoroughly investigated. To address this research issue, in this paper we discuss one of the fundamental elements that is designed to facilitate as well as to provide tool support for inter-organizational ITSM processes - *ioCMDB*.

Our discussion is motivated by an inter-organizational IT management scenario in the Géant project, which is a pan-European collaboration on high-performance networks as the e-Infrastructure to support international research cooperation in areas such as high-energy physics. To provide sufficient capacity for communication of data deluge generated by scientific experiments, dedicated E2E Links have to be provisioned across different organizations in order to transport huge amount of research data in a timely manner.

In this paper, we show the challenges of managing complete life cycle of such E2E Links across various management domains. To tackle such challenges, a management tool called

I-SHARe is designed and developed for Géant specifically. The goal of I-SHARe is to provide a centralized repository of management information of links throughout their life-cycle. Any changes of a link, such as changes of routes, should be reflected accordingly and information must be updated in a timely manner. In our previous work, a holistic reference model is derived with the intention to generate new artifacts or evaluate an existing artifact. In this paper, however, we concentrate on the evaluation of the I-SHARe in accordance to the ioCMDB reference model. The ultimate goal of this work are manifold: on one hand, the ioCMDB generated reference model are applied as an auxiliary mean to identify the potential improvement opportunities to make I-SHARe a better tool for inter-organizational management; on the other hand, the practical aspect of an ioCMDB approach is put under tests in a real-world situation.

Through a careful evaluation and optimization, improvement opportunities for the current version of I-SHARe are identified consequently. As our future work, these achieved results will be reflected in the next version and will furthermore influence our future design of I-SHARe. With considerations of the identified improvement potentials, I-SHARe will gradually evolve into a tool that could serve generic inter-domain e-Infrastructure management purposes rather than being specially tailored for a set of rather limited use cases.

Acknowledgment The authors thank the members of the Munich Network Management Team for valuable comments on previous versions of this paper. The MNM-Team, directed by Prof. Dr. Heinz-Gerd Hegering and Prof. Dr. Dieter Kranzlmüller, is a group of researchers from the University of Munich, the Technische Universität München, the German Federal Armed Forces University in Munich, and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. The web server of the MNM Team is located at <http://www.mnm-team.org/>.

REFERENCES

- [1] M. Yampolskiy, W. Fritz, W. Hommel, G. Vuagnin, and F. Galeazzi, "Manual multi-domain routing for géant e2e links with the i-share tool," in *INTERNET 2011, The Third International Conference on Evolving Internet*, 2011, pp. 98–103.
- [2] A. R. Hevner, "The three cycle view of design science research," *Scandinavian Journal of Information Systems*, vol. 19, no. 2, pp. 87–92, 2007.
- [3] S. Knittl and K. Beronov, "E-Infrastructure Projects from a Bavarian Perspective: Potentials of Standardization," in *eChallenges 2011*, Florence, Italy, 2011.
- [4] "European Commission - ICT research in FP7 - e-Infrastructure," available online at http://cordis.europa.eu/fp7/ict/e-infrastructure/projects_en.html, access on 2011-09-12.
- [5] S. Knittl, "Werkzeugunterstützung für interorganisationales IT-Service-Management – ein Referenzmodell für die Erstellung einer ioCMDB," Ph.D. dissertation, Technische Universität München, 2012.
- [6] J. Knobloch and L. Robertson, "LHC computing Grid, Technical design report," CERN, <http://lcg.web.cern.ch/LCG/tdr/>, Tech. Rep., 2006.
- [7] "WLCG website," <http://lcg.web.cern.ch/LCG/public/>, 2011.
- [8] K. Schauerhammer and K. Ullmann, "Operational Model for E2E links in the NREN/GÉANT2 and NREN/Cross-Border-Fibre supplied optical platform," Géant, Tech. Rep., 2006.
- [9] P. Marcu, D. Schmitz, W. Fritz, M. Yampolskiy, and W. Hommel, "Integrated monitoring of multi-domain backbone connections—operational experience in the lhc optical private network," *Arxiv preprint arXiv:1101.3896*, 2011.

- [10] G. Cesaroni, M. Hamm, F. Simon, G. Vuagnin, M. Yampolskiy, M. Labeledzki, and M. Wolski, "I-SHARE: Prototype specification," Géant, Tech. Rep., 2008.
- [11] "I-SHARE pilot installation," cs.ishare.geant.net/, 2011.
- [12] M. Yampolskiy, G. Vuagnin, and M. Łabędzki, "I-SHARE: End-of-Pilot Review and Future Steps," GÉANT, Tech. Rep., 2011.
- [13] M. Yampolskiy and M. Hamm, "Management of multidomain end-to-end links; a federated approach for the pan-european research network géant 2," in *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*. IEEE, 2007, pp. 189–198.
- [14] OGC, *ITIL V3 complete suite - Lifecycle Publication Suite*. The Stationery Office Ltd, 2007.
- [15] W. Hommel, S. Knittl, and D. Pluta, "Availability and Continuity Management at Technische Universität München and the Leibniz Supercomputing Centre," in *15th International Conference of European University Information Systems (EUNIS 2009)*, Santiago de Compostella, Spanien, Juni 2009.
- [16] P. Fettke and P. Loos, *Reference Modeling for Business Systems Analysis*. IGI Publishing, 2006.
- [17] S. Knittl and M. Brenner, "Towards a configuration management system for hybrid cloud deployments," in *6th IFIP/IEEE International Workshop on Business-driven IT Management (BDIM 2011)*, Dublin, Irland, Juni 2011.
- [18] H.-G. Hegering, S. Abeck, and B. Neumair, *Integrated Management of Networked Systems - Concepts, Architectures and their Operational Application*. Morgan Kaufmann Publishers, 1999.
- [19] P. Marcu and W. Hommel, "Inter-organizational fault management: Functional and organizational core aspects of management architectures," *CoRR*, vol. abs/1101.3891, 2011.
- [20] S. Knittl, T. Schaaf, and I. Saverchenko, "Change management in e-infrastructures to support service level agreements," in *Managing and Delivering Grid Services (MDGS 2011)*, Bordeaux, Frankreich, August 2011.
- [21] M. Yampolskiy, W. Hommel, P. Marcu, and M. K. Hamm, "An information model for the provisioning of network connections enabling customer-specific end-to-end qos guarantees," in *IEEE International Conference on Services Computing (SCC)*, Miami, FL, July 2010, pp. 138 – 145.
- [22] A. J. Gilliland-Swetland, "Setting the stage: Defining metadata," in *Introduction to Metadata: Pathways to Digital Information*, M. Baca, Ed. Getty Publications, 2008. [Online]. Available: <http://nsl.nisclair.res.in/bitstream/123456789/954/1/Introduction+to+Metadata.pdf>
- [23] L. Brodner, "Visualization of management information in cloud computing," Thesis, Ludwig-Maximilians-Universität München, 2011.
- [24] W. Hommel and S. Knittl, "An access control solution for the inter-organizational use of itil federated configuration management databases," in *Workshop of HP Software University Association (HP-SUA)*, 2008.
- [25] H. von Jouanne-Diedrich, J. Blechinger, C. P. Neumann, S. Schwarz, and R. Lenz, "Integration verteilter und heterogener Configuration-Management-Datenbanken - Eine Anforderungs- und Marktanalyse," *Informatik Spektrum*, vol. 33, no. 4, pp. 351–362, 2010.
- [26] H. Yamada, T. Yada, and H. Nomura, "Developing network configuration management database system and its application - data federation for network management," *Telecommunication Systems*, pp. 1–8, September 2011. [Online]. Available: <http://dx.doi.org/10.1007/s11235-011-9607-0>
- [27] F. Carlisle, B. Day, S. Donohoo, S. Dublisch, and et. al., "Cmdb federation (cmdbf) - committee draft - version 1.0b," Website, Januar 2008. [Online]. Available: <http://cmdbf.org/>
- [28] D. Scott, D. Williams, and R. J. Colville, "The future of it operations management suites," Gartner, Tech. Rep. ID:G00163656, Januar 2009.
- [29] R. J. Colville and G. Spafford, "Four pitfalls of the configuration management database and configuration management system," Gartner, Tech. Rep. ID:G00210078, Januar 2011.
- [30] L. Pasquale, J. Laredo, H. Ludwig, K. Bhattacharya, and B. Wassermann, "Distributed cross-domain configuration management," in *Service-Oriented Computing*, ser. Lecture Notes in Computer Science, L. Baresi, C.-H. Chi, and J. Suzuki, Eds. Springer Berlin / Heidelberg, 2009, vol. 5900, pp. 622–636.
- [31] H. Kozankiewicz, K. Stencel, and K. Subieta, "Implementation of federated databases through updatable views," in *Advances in Grid Computing - EGC 2005*, ser. Lecture Notes in Computer Science, P. Sloot, A. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, Eds. Springer Berlin / Heidelberg, 2005, vol. 3470, pp. 393–396.
- [32] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*. Springer, 2011.
- [33] A. Bauer and H. Günzel, *Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung*, 3rd ed. dpunkt, 2009.

Introducing openBOXware for Android: The Convergence between Mobile Devices and Set-Top Boxes

Lorenz Klopfenstein^{1,2} Saverio Delpriori^{1,2} Gioele Luchetti^{1,2}

Andrea Seraghiti¹ Emanuele Lattanzi^{1,2} Alessandro Bogliolo^{1,2}

¹STI-DiSBef - University of Urbino, Urbino, Italy 61029

²NeuNet Cultural Association, Urbino, Italy 61029

E-mail: lck@klopfenstein.net saveriodelpriori@gmail.com luchetti@sti.uniurb.it
andrea.seraghiti@uniurb.it emanuele.lattanzi@uniurb.it alessandro.bogliolo@uniurb.it

Abstract—Multimedia contents delivered over residential and mobile IP networks are among the main driving forces of the Internet. The pervasiveness of connected devices capable of receiving and decoding multimedia streams has induced a change in the market of set-top boxes from dedicated proprietary appliances to software modules running on top of off-the-shelf devices. In spite of the large number of devices we use every day, smartphones are the favorite answer to our communication needs because of their availability, of their user friendliness, and of the great opportunities of personalization offered by user-generated mobile applications. The last generation of smartphones and tablet PCs, capable of handling HD multimedia streams while also retaining the distinguishing features of mobile devices, enables the convergence between personal communication devices and home entertainment appliances. This paper introduces openBOXware for Android, an application suite which makes it possible to use any Android device as a set-top box, allowing end-users to take advantage of the tailored run-time environment of their personal mobile devices while watching television in the comfort of their living rooms. OpenBOXware exploits technology convergence for usability, in the attempt of enhancing the accessibility of the Internet by providing a TV-like usage experience. The paper presents the key features of openBOXware, outlines the implementation on top of the Android application framework, and shows representative use cases.

Keywords—Set-top box, Tablet PC, openBOXware, Android, Streaming

I. INTRODUCTION

The analog switch-off and the advent of *digital video broadcasting* (DVB) have enabled the technological convergence of client-side equipment required to take advantage of broadcast TV channels, IPTV services, and Internet multimedia streams. Nowadays, all new television sets come with embedded decoders, and most of them are Internet enabled. In this scenario, software components running on top of off-the-shelf connected devices are replacing proprietary *set-top boxes* (STBs), while traditional IPTV models are undergoing deep changes in order to face the pressure of *over-the-top* (OTT) multimedia contents streamed across global *content delivery networks* (CDNs).

At the same time, the widespread diffusion of smartphones and Internet enabled mobile devices, together with

the growing coverage of broadband wireless networks, have induced operators to move from *triple-play* offers (i.e., Internet access, VoIP, and IPTV) to *quadruple-play* offers (which include mobility) [2], accelerating the convergence between mobile and residential broadband markets and creating the conditions for delivering mobile TV services [3]. IP traffic trends and forecasts [4], [5] indicate that multimedia contents delivered over residential and mobile IP networks are among the main driving forces of next generation networks.

In spite of the wide diversity of connected devices which might work as multimedia boxes (including connected TV sets, media centers, DVB decoders, video game consoles, and personal computers), end-users spend most of their connected time using personal smartphones (or similar handheld devices) which have several competitive advantages: they are available everywhere and at any time, they offer intuitive user interfaces, they provide suitable answers to any communication need, and they provide unprecedented opportunities of personalization thanks to the thriving market of user-generated contents and applications [6].

Exploiting add-ins and configuration options to create a perfectly tailored run time environment on a smart phone is an intriguing pastime that engages the vast majority of end-users. As a result, both the quality of experience offered by smartphones and the effort devoted to personalize them keep end-users from using (or at least from personalizing) other devices.

Although a new generation of STBs has recently sprouted which allow end-users to create their own applications and to easily install third-party addins [7], [8], they are far away from gaining the popularity of their mobile counterparts and the gap is hard to be closed in the near future. In fact, mobile devices are always at users' disposal and they will maintain their dominant role of personal communication equipment. Moreover, STBs are typically installed in a living room where they are mainly expected to provide a *lean-back* usage experience, which is very well suited for media consumption and is in contrast with the *lean-forward* attitude typical of smart phone users, which has sustained the market of mobile

applications [9], [10].

On the other hand, personal handheld devices have never threatened the market of media centers and STBs because of their tight design constraints imposed by portability requirements, which made them unsuitable to sustain the workload of high definition multimedia streams. The gap between personal mobile devices and multimedia boxes has been closed, however, by the last generation of smartphones and tablet PCs, which support HD video streams and are equipped with HDMI interfaces, and by the advent of IP boxes providing the same application framework of the most popular mobile devices [11].

In a preliminary version of this paper [1], the authors investigated the possibility of making an Android tablet PC work as a STB in order to allow end-users to take advantage of their personal runtime environment in the comfort of their living room. This paper moves a step forward by outlining the key features and the implementation details of openBOXware (OBW) for Android, a modular application suite which makes it possible for Android devices, including smartphones, to switch from a *lean-forward* to a *lean-back* usage mode in order to provide a TV-like experience of Internet contents. The main purpose of openBOXware is to exploit this convergence, making both the advanced features of Android and the unlimited contents available on the Internet directly accessible to television viewers, presenting contents in a familiar way: as linear TV channels, possibly controlled with a simple remote control.

The rest of the paper is organized as follows: Section II presents the concept and the main features of openBOXware, Section III outlines the software architecture and the implementation details, Section IV shows representative use cases, and Section V draws conclusions.

II. OPENBOXWARE FEATURES

OpenBOXware is an open-source framework built on top of Android to provide a TV-like experience of multimedia contents taken from heterogeneous sources (in terms of format, protocol and access mode), while also allowing the end-user to enjoy all the applications installed on the underlying Android device. To this purpose openBOXware sports a custom user interface conceived to offer a lean-back usage experience by means of three home screens, granting access to: the *media library* (Figure 1), the list of *openBOXware applications* (Figure 2), and the list of all other *Android applications* installed on the device.

The media library is the default home screen, which allows the end-user to find media channels and to select the one to watch. Multimedia contents are made available by special add-ins, called *media sources*. A media source is a tree of nested multimedia nodes. Leaf nodes are *playable*, in that they can be forwarded to the media player for playback, while all other nodes are *explorable*, in that they allow the media library to navigate their content and display the

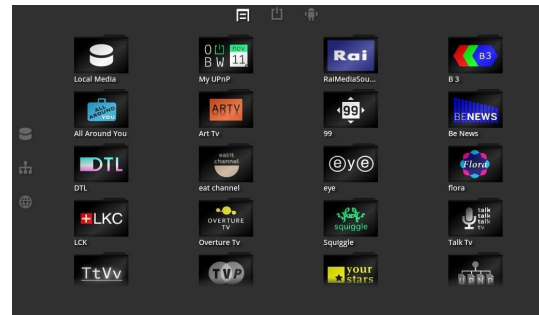


Figure 1. Media library home screen.

list of children nodes. Examples of media sources include IPTV channels, Internet TV channels, UPnP/DLNA clients granting access to the multimedia contents made available by the UPnP/DLNA servers discovered in the LAN, collections of media elements stored in the local file system, and collections of online multimedia contents.

Media source nodes may contain metadata, including title, duration, and an icon, that can be displayed by the media library to provide a richer and more vivid browsing experience and to help the end-user to decide which content to pick.

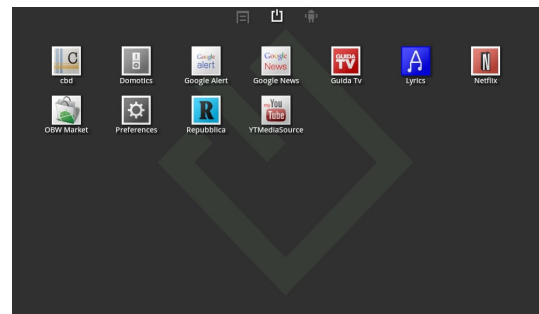


Figure 2. OpenBOXware applications home screen.

Figure 1 shows the media library home screen, with the icons of all the media sources installed in the device. When a media source is selected, the media library shows the icons of its children nodes. The three small icons on the top of the screen can be used to switch among the three homes, while the ones on the left represent filters that can be applied to the media sources based on the location of the contents they link to: local file system, LAN, Internet.

Playable media source nodes provide a TV-like watching experience by offering both linear channels or contents on demand. A content on demand is a media source node associated with a single resource which is played back whenever the media source node is selected by the user. The node's content do not change and depend exclusively on the user's choice. A linear channel, on the contrary, can be either a link to a continuous stream provided by a live streaming server, or a (possibly unlimited) list of disjoint multimedia



Figure 3. OpenBOXware media player with visible control bar.

elements which are glued together by the node and played back as a continuous stream. In this case, contents may depend on the moment in time the user decides to tune in.

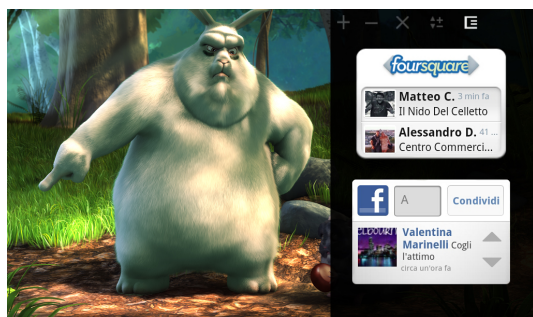


Figure 4. Sidebar displayed over the media player.

OpenBOXware applications are special Android applications which make use of additional features provided by the openBOXware library included in the Software Development Kit (SDK). Applications can be executed in *fullscreen* (if they take over the whole screen area, covering up other applications), in *background* (in case of services that do not require any graphic user interface), or in *sidebar* (the case of widgets that can be displayed on a small part of the screen letting the top-level fullscreen application shine through). An example of sidebar applications displayed on top of the media player is provided in Figure 4.

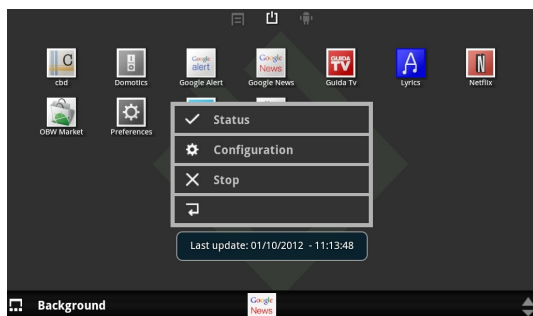


Figure 5. Background application management dialog box raised from the control bar.

To support a lean-back usage experience, openBOXware provides an overlay interface element, called *control bar*, that is displayed at the bottom of the screen with minimum interference with the foreground activity. User controls are organized in sections which differ in type and scope. Each section takes the entire area of the control bar, so that the control sections are displayed one at the time. This is due to two main reasons: first, to limit the area of the screen taken by the control bar, second, to make it easier for the end-user to issue commands which are limited in scope. In particular, both the switching among the sections and the controls contained in each one can be operated by means of the four arrows and the “OK” button available in any remote control. Control bar sections include: *playback* (Figure 3), containing a seek bar together with pause, skip, and stop buttons; *volume*, which controls the volume settings of the device; *home*, which provides short cuts to the three home screens; *sidebar*, which controls the configuration of the sidebar and the widgets displayed in it; and *background*, which provides a scrollable list of services running in background and allows the end-user to pick one in order to check its status, change its configuration, or stop it (Figure 5).



Figure 6. Overlay digits providing feedback of the zapping command issued while watching a movie.

To further improve usability, any playable media element can be associated with a unique 3-digit number to be used as a short cut to directly zap to that channel from the media player (or from any home screen) without going through the media library and browsing media sources. Figure 6 shows the three digits which appear in overlay whenever the end-user presses a numeric key in the remote control to exploit the zapping functionality.

OpenBOXware also supports the so called *configurable media sources*: essentially standard media sources bundled with companion openBOXware applications that can be used to configure them. In what ways and to what extent a media source can be customized depends on the structure of the multimedia provider for which the media source is developed, but the auxiliary application typically provides a user interface to change settings and preferences, to set search criteria, or to apply filters. Examples of configurable media sources include a UPnP client with an auxiliary

application to be used to associate a channel to a specific directory or to a specific file made available by some UPnP server in the LAN, or a YouTube media source with an auxiliary application allowing the end-user to create a channel associated with a specific query and search options. In both cases, the channels created by the end-user through the companion applications appear as new children of the corresponding media sources and they can be used as any other playable node and possibly associated with numeric shortcuts for quick zapping.

This advanced feature grants to openBOXware the flexibility required to provide a TV-like experience of any online contents (even if they are not natively organized in linear channels) making them available to a broader audience. For instance, people not accustomed to browse the Internet can take advantage of thematic channels preinstalled or created by other expert users. In addition, parent can create channels of contents expressly selected and filtered for their children.

III. OPENBOXWARE IMPLEMENTATION

OpenBOXware is built on top of Android which is, in its turn, built on top of Linux kernel. The Android architecture consists of three main layers: *i)* the *Android runtime*, based on the *Dalvik virtual machine (VM)* with additional support libraries, *ii)* the *application framework*, and *iii)* the *applications* which run on it [11]. For portability and compatibility reasons, openBOXware has been fully developed at the application level.

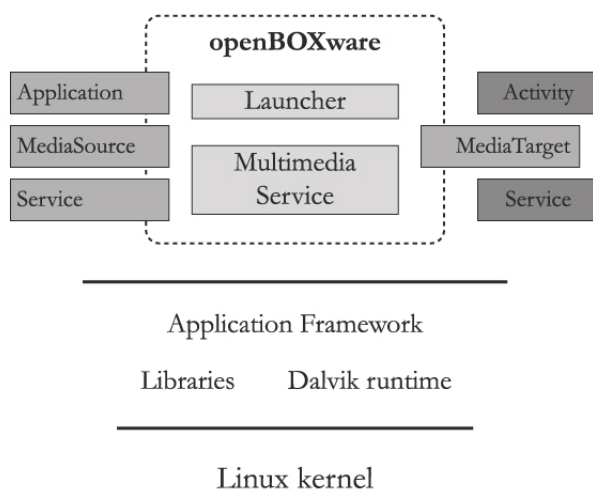


Figure 7. Software architecture of openBOXware.

An Android application can be made of several components. For our purposes, the most important types of components are *activities*, which represent screens with specific user interfaces, and *services*, which run in background. Each application runs in a separate VM instance for security and protection. Communication among applications is guaranteed by an asynchronous message passing mechanism which

allows a component to issue an *intent* message which is handled by another component, possibly belonging to a different application. Each intent contains action and data specifications which are used by the application framework to dispatch the intent and trigger one of the components registered for performing the requested action on the specific type of data. The main graphic user interface is provided by a *launcher*, which is a special activity registered to react to a particular intent issued by the operating system at start up. The launcher allows the end-user to browse and launch activities which publish the *MAIN* intent filter. In addition, the launcher can also act as a *widget host* to allow end-users to customize the main page by embedding their preferred miniature application views.

The *openBOXware core* is implemented as a package containing a launcher and several additional components, including background services and other activities, which handle multimedia functionalities, notifications, and sidebar widgets. As mentioned in Section II, the user interface of openBOXware discriminates between normal Android applications and special openBOXware applications (identified by the intents they are registered to handle, as detailed below). An openBOXware application is nothing more than a standard Android application, mainly composed of activities and services, which additionally relies on the APIs exposed by the *openBOXware SDK* and links to the *openBOXware library*. Those APIs, written on top of the features provided by the *vanilla Android* platform, enable the application to be integrated inside the openBOXware environment and give access to advanced multimedia features through high level programming interfaces.

A schematic representation of the software architecture is provided in Figure 7, where the openBOXware platform rests on the Android application framework. Media sources and openBOXware applications are represented as boxes partially overlapping the openBOXware core to denote the fact that they make use of the extended features provided by the Software Development Kit, while all other Android applications (activities and services) do not.

A. Home application

The core package, once installed onto an Android device, works as a launcher which provides the three home screens from which media sources can be explored and any other application is launched.

It is worth noticing that multiple launchers can be installed on the same device, but only one at the time can be set as default and run in foreground. Hence, the device must be setup to allow the user to select the launcher in order to switch from a *lean-forward* to a *lean-back* use of his/her own device. This can be achieved either by changing the default launcher, or by avoiding to specify a default one (in this case the choice is made every time the end-user taps the home button, through an Android dialog box – shown

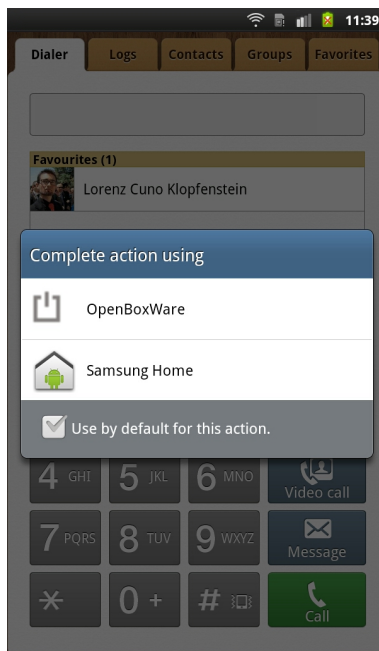


Figure 8. Launcher selection dialog box.

in Figure 8 for a Samsung smartphone – which allows the user to pick the desired launcher).

Once the *openBOXware* launcher is launched, it becomes the main graphical user interface of the system, acting as an application launcher and displaying additional messages through the system bar and a custom control bar. As mentioned in the previous section, the launcher provides three separate home screens: the media library, the *openBOXware* application list, and the Android application list.

B. Media source

A media source is implemented as an Android service, reacting to the `openboxware.media.source.ACCESS` intent. The media library (and in fact any other Android application) can search for media sources by querying for that specific intent and getting a list of the installed services. Media sources may use an additional category (namely `openboxware.media.source.LOCAL`, `LAN`, or `INTERNET`) to specify the location of the data the media source points to. These are known as content-type flags. The Android package manager can filter out specific kinds of media sources based on the location specified in the intent query. From a user-interface perspective, filters are activated by the three clicking on the three icons available on the left-hand side of the media library home screen (see Figure 1).

Media sources implement a set of remote procedure calls (RPCs) using the Android *AIDL* interface. The *openBOXware* library provided by the SDK includes a collection of interfaces and higher level classes (contained in the *Tahweed* APIs) that make implementing such a RPC interface much

easier and more structured. The developer can simply implement a small set of classes and connect the exploration actions (i.e., their RPCs) to the corresponding back-end code needed to access the specific media content.

Each node of a media source is an instance of the *MediaSourceNode* class, which provides a set of abstract methods that need to be overridden in order to be used by the client to fetch children nodes and multimedia resources.

Playable media source nodes provide an enumerable sequence of content items, which will be played one after the other by the media player. Each item of this stream is an instance of the *MediaElement* class and can link to any remote or local multimedia resource (including images, audio files, and videos) through its URI.

Since every playable node includes a – possibly unlimited – stream of multimedia elements, as mentioned in Section II they can be used to create a linearized channel. An example thereof can be a virtual TV channel built from multiple separate videos, possibly encoded in different formats and taken from different sources. Media source nodes can also transform any feed of data into a linearized stream of media. For instance, any feed containing links to images (an RSS feed, possibly) can be represented as a stream of pictures, which is then played back by the media player as a slideshow.

C. Media library

The media library is the default home screen of the *openBOXware* launcher, which represents the main user interface to the platform's multimedia exploring and streaming capabilities: its primary function is to list all media sources installed and available to the user, allowing him/her to explore media sources' contents, discover media channels, and select the ones to watch.

Media sources are displayed in a grid in alphabetical order, as shown in Figure 1. When the user decides to select one particular media source, the *openBOXware* library opens an *AIDL* connection to the respective service and starts communicating with the service implementing the media source (through the RPC interface described before). Media source's nodes are explored hierarchically: the history of the exploration is kept in a stack and displayed on the left-hand side of the media library grid, enabling the user to easily understand his/her position in the media source's structure and to navigate back either by using the *back* button or by clicking on the stacked ancestral nodes.

When the user clicks on the icon of an explorable node the media library pushes it into the stack and displays its children nodes. When he/she clicks on a playable node it is passed on to the integrated media player, which attempts to play back the contents of the node starting from the first multimedia element.

Finally, the media library includes a link to the Android marketplace with a shortcut that quickly filters out installable

media sources: the user can explore applications which are available to download, install them, and get back to the media library to make use of the new media sources and of the multimedia resources they provide access to.

D. Media player

The media player (Figure 3) is the openBOXware component that handles all multimedia playback requests by the media library (and by other openBOXware applications): it is implemented as a single activity sporting a simple user interface and capable of playing back a variety of linearized contents as described by media source nodes (instances of *MediaSourceNode*) exposed by the media sources installed on the device.

The player activity reacts to the `openboxware.mediaplayer.PLAY` intent, which signals the request for playback issued by another application. The intent structure contains all the data required by the media player to initiate the playback. In case of media source nodes containing a single media element, the intent contains directly the *MediaElement* instance to be played back. In case of playable media source nodes containing more elements, the intent contains a so-called *media source node identifier*, which will be used by the media player to open a connection to the original media source, make a request for the node to playback via RPC, and then start enumerating the multimedia resources provided by the node. Each single media element returned by the node is reproduced as part of a continuous linear stream.

The media player determines the type of each media element to play back. Videos and audio files are played back using the default Android *MediaPlayer* component, while pictures are displayed by a custom slideshow component. Media identification relies on the metadata provided by the developer of the media source, which include a MIME type specification (i.e., a string formatted according to the *Multipurpose Internet Mail Extensions* standard, which gives a textual description of the content type of a file, such as “text/html” or “video/x-h264”). If absent, the media player will attempt to guess the nature of the media element (e.g., by checking the file extension in the element’s URI or by other heuristics). When an unsupported (or unknown) media element is passed to the media player, playback fails and the media player attempts to skip it and go to the next element of the media source node.

The media player activity does not display any additional user interface elements, nor any buttons that enable the user to control media playback. Instead, the player emits a sequence of intents to the system notifying its current status, time of playback, and other additional data (like the name of the current media element which is being played back). These data intents can be intercepted by any other component of the system to keep track of the

player’s actions. Most notably these intents are used by the control bar which, when shown, displays playback progress, common playback controls, and other data directly to the user (as shown in Figure 3).

On the other hand, the media player can also be controlled by generating special intents which represent commands to pause, resume, skip, or stop playback. By default, these intents are sent by the control bar when the user clicks on the corresponding control buttons, but they can also be used by other applications and services that might need to interact with the media player.

E. Sidebar

The sidebar (shown in Figure 4) is implemented as a single instance activity which is displayed as a transparent dialog, covering only a small side of the screen. Because of limitations of the Android window manager, dialog activities which do not cover the whole screen take the entire input focus of the user. Thus, interaction with the underlying fullscreen activity is either impossible or unreliable because of its dependence on device-specific implementation choices. This problem is partially mitigated by the fact that the most common fullscreen activity, i.e., the media player, does not provide any touch interface and all interactions usually pass through either the remote control (via intents) or the control bar (which is never displayed together with the sidebar).

The sidebar activity acts as a so-called *widget host*, which can accommodate any number of external widgets implemented by other applications installed on the system. Widgets commonly used on Android devices include simple front-ends to communication applications (e-mail readers, contacts lists, incoming text message viewer, feed readers, ...) and social networking clients (Twitter, Facebook, LinkedIn, ...).

Widgets added to the sidebar are identified by a unique *appWidgetId* that is used to grant persistence to the list of hosted widgets, to store their display order as decided by the end-user, and to handle removal. Widgets are automatically updated by the Android system.

The sidebar provides simple control buttons that can be used to add, move, or remove widgets either through the touch screen or from a remote control.

F. OpenBOXware applications

Different classes of applications can be developed to target the openBOXware environment, in order to make use of the features of the framework or to integrate with the media library. In particular, openBOXware supports four ways of integrating third party add-ins.

Interactive applications (namely, *fullscreen* openBOXware applications) can be implemented as activities that will run taking the whole device screen over and delivering an immersive usage experience to the user. As mentioned in Section II, these applications run one at a time, demand

exclusive focus from the user and can also rely on the media capabilities of openBOXware to explore media resources or demand media playback. Applications of this class will be listed separately in the openBOXware applications list by the launcher (they will be hidden from the standard Android applications list of the openBOXware launcher and of other Android launchers, by default).

Background applications, which usually play the ancillary role of helper services to a fullscreen activity in order either to periodically fetch updated data or to poll for some resource, can be implemented as Android services. These applications will also appear in the openBOXware applications list and can be explicitly launched in background by the user. The control bar will also provide a graphical interface that allows the user to interact with such background services, updating their configuration and eventually terminating them. A background application can interact with the openBOXware framework, explore media resources, and invoke other fullscreen activities when needed.

Other applications that do not request the full focus of the user and allow to glance at information without interrupting the main usage experience can be implemented as standard Android widgets. Widgets have limitations on how often their code runs and how their interface is displayed, but they can be included in so-called host applications. Creating a widget allows the application to be hosted by the openBOXware *sidebar*, enabling the user to display the application while interacting with another fullscreen application (for instance, the media player). Every widget can be hosted by any widget host, for instance other Android launchers that provide this functionality.

Finally, it is worth mentioning that *media sources* are application as well. In particular, they are implemented like standard Android services, but they implement a specific interface that enables the media library (and in fact any other application written using the openBOXware SDK) to access the service and fetch informations about the available media resources from the media source's hierarchical resource tree.

Being implemented as standard Android activities and services, openBOXware applications are able to react to common Android intents. It is up to the developer to include the default Android intents in their applications to enable them to be launched through any Android launcher instead of relying only on the openBOXware front-end. It is worth noticing, however, that in this case the openBOXware services might not be available.

In order to be listed and launched as openBOXware applications, activities must handle the `openboxware.gui.FULLSCREEN` intent and should extend the `FullscreenActivity` class included in the SDK. This allows the application to easily access all openBOXware features (e.g., raising the control bar, displaying the sidebar, issuing commands to the media player, or raising notifications, as described in Section

III-H3). Fullscreen activities also automatically handle their own theme when launched, in order to match the look of the platform and the size of the screen without any additional effort for the developer.

Similarly, background services must handle the `openboxware.gui.BACKGROUND` intent in order to be listed among the openBOXware applications, and they have to extend the `BackgroundActivity` class to gain access to openBOXware features. In particular, this allows them to be launched, monitored, configured, and stopped by the openBOXware control bar.

As already mentioned, media sources handle the `openboxware.media.source.ACCESS` intent and extend the `MediaSource` class, which implements the basic AIDL which allows media source clients (like the media library) to discover multimedia content through RPC calls to the service. Media sources can also rely on the high-level *Tahweed* multimedia API, which provides a structured object-oriented wrapper around the imperative RPC constructs on which the Android inter-process communication system is based.

While all Android applications have access to platform's *context* in order to query base services and managers, openBOXware applications can also access their `OBWApplicationContext`, which enables them to integrate with the features of the platform and provides a mean to easily share status and information between different activities/services which are part of the same application. In particular, this allows the developer to implement an openBOXware application that supports all execution modes (i.e., fullscreen, background, sidebar) and switches among them without losing data and settings.

G. Control event injection

The control bar is implemented as a standard Activity and communicates with the openBOXware back-end through intents (e.g., as already mentioned, the media player signals its state continuously using intents and can be controlled by broadcasting other specific intents representing user's commands). The current implementation of the control bar allows the user to interact with the openBOXware environment both in a *lean-forward* stance, by using the touch interface of the device, and in a *lean-back* stance, by using a remote control to issue commands. Any Bluetooth remote that is able to connect to the Android device can be used to this purpose.

In order to maintain full compatibility with existing Android devices, the decision was made not to implement a custom Android build with system-wide changes. This hinders the capability of openBOXware to provide deep integration with a remote control promoted to a full-fledged interaction device capable of injecting interface events to Android applications other than the openBOXware launcher. On the other hand, this kind of customization could be

provided in device-specific openBOXware distributions targeting Android IP boxes.

H. Advanced features

1) *Configurable media sources*: Configurable media sources are standard media sources that include a companion openBOXware application. The media source and the configuration application share the same Android application package and thus also share a common space of isolated storage on the device. This storage area is used to store structured information (usually a SQLite database, a shared preference structure, or simple files in any parsable format) that can be altered by the application and read by the media source in order to display custom contents based on queries and preferences specified by the end-user.

A representative example of a configurable media source is shown in Section IV.

2) *Zapping*: Any playable media source node can be marked as *pinnable* in order to enable the zapping functionality. In practice, this means that the media library is allowed to store a reference to that particular node, i.e., to *pin* it and associate it to a number. OpenBOXware lets the end-user create a numbered set of preferred channels, which he/she can play back quickly and easily by *zapping* to the corresponding numbers. On the other hand, a pinnable node requires the ancestral media source to be able to regenerate that node and its complete state (which could require a complex interaction with the back-end content provider). This is done by generating an identifying code for the pinned node (i.e., an arbitrarily complex string constructed by the media source and containing all relevant information) that can be parsed at any time (even on a different instance of the same media source or after a full device reboot) in order to regenerate the original media source node.

In practice, the launcher makes use of the zapping function by allowing the user to pin nodes when they are declared as pinnable by the media source developer. When the user executes a long-press on a pinnable node displayed in the media library (holding the node for a couple of seconds), the library displays an overlay that enables the user to select a channel number for that particular node. Once the user confirms the input, the node is pinned and associated with that number.

Subsequently, whenever the end-user picks the channel number using either the device or the numeric keys of the remote control, the media library attempts to retrieve the corresponding pinned node and use its identifying code to restore the desired `MediaSourceNode` instance. The node is then forwarded to the media player and directly played back.

3) *Notifications*: Standard Android notifications can be sent to the Android notification bar by any application by instantiating a new notification and sending it to the `NotificationManager`, which is part of the standard Android

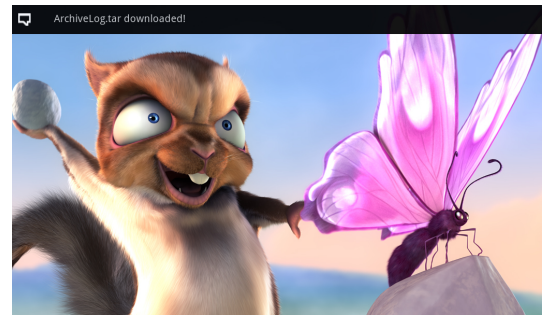


Figure 9. Example of an openBOXware notification.

programming interface. The notification is then displayed in the notification bar until it is dismissed by the user.

Since the Android notification bar is covered by the openBOXware launcher by default, these notifications are hidden to the user and cannot be easily accessed. While any application can still make use of the default notifications system, raised notifications will only be displayed once the openBOXware launcher is terminated. If, on one hand, this behavior is intentional in that it avoids notifications, which usually interrupt the normal usage experience by prompting for user's attention, to detract from the *lean-back* usage experience typical of openBOXware, on the other hand, a less-obtrusive form of notifications are needed, to be possibly consumed while watching multimedia contents.

OpenBOXware notifications can be displayed by applications using the openBOXware APIs included both in the `FullscreenActivity` and in the `BackgroundActivity` base classes, instead of relying on the default Android notification manager.

The notification request is taken over by the openBOXware environment, which will display it by using a simple notification overlay on one edge of the screen (see Figure 9), while also forwarding the same notification to the underlying default `NotificationManager` to ensure that the notification can be read and acknowledged later, even if ignored while on screen.

When the user reacts to an openBOXware notification (by clicking or touching the overlay icon while displayed), the system hides both the overlay notification and the corresponding entry in the standard Android notification bar. At the same time the notification raises a custom intent that can be handled by the application.

IV. RELEASE

A. Public demonstration

A pre-alpha version of the openBOXware framework was tested and publicly demonstrated on November 11, 2011 in Urbino (Italy), using a living room setting installed in a conference hall at the University of Urbino.

A HD TV set was connected to a Samsung Galaxy SII smartphone running openBOXware, providing a replica of

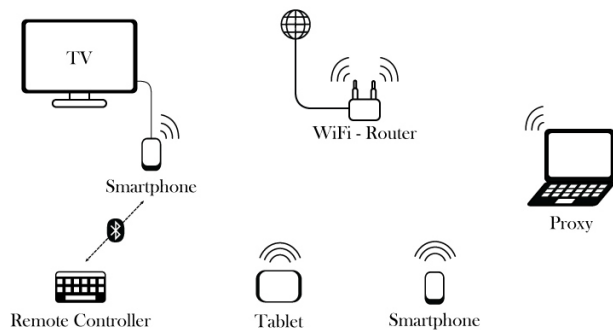


Figure 10. The openBOXware test bed, including a TV set driven by an Android smartphone controlled by a Bluetooth remote and several other Android devices connected to the same WLAN.

the screen user interface (in its original display resolution) through the HDMI port. The smartphone was placed close to the TV and far away from the sofa in order to force the end-user to rely on a remote control to issue commands (namely, a Bluetooth *Logitech diNovo Mini wireless keyboard* that can work both as a remote and as a wireless keyboard/mouse input device). The smartphone was connected to the Internet through a Wi-Fi router. Other Android devices (smartphones, tablet PCs, and IP boxes) running openBOXware were connected to the same WLAN for testing purposes. One of the smartphones was also running an UPnP server in the background. All devices were able to access Internet resources, files shared by the UPnP server, and resources stored on a HTTP server installed on a local computer playing the role of a proxy of video contents from RAI Radiotelevisione Italiana.

A schematic representation of the setting is provided in Figure 10. The setup was used to demonstrate the usability of openBOXware in a pure *lean-back* setting, to test the multimedia playback capabilities of the devices involved, and to show the key features of the platform by means of representative use cases. A video log of a demo is available at: <http://youtube.com/watch?v=7RzdIiz1EQs>.

Demonstrated features include: home screen navigation (see video log at 2:49s), media source exploration of representative media sources such as a Youtube channel, a UPnP client, and a HTTP proxy of RAI streaming contents (at 4:18s), playback of a full HD video, control bar (at 5:28s, with event injection from both the touch screen and the remote control), sidebar (5:57s), channel shortcut assignment and zapping (6:55s), multiple launch modes (8:10s) shown on a simple Google News client supporting both full screen and background execution modes, and configurable media source samples, such as UPnP and Youtube clients (10:20s).

The experiments conducted on configurable media sources are worth to be described in detail. Both the YouTube and the UPnP media sources developed for testing purposes had their own companion applications to be used to customize them

outside the media library (which in fact provides only an interface for content consumption). The UPnP configuration application (called “My UPnP”) showed the list of all available UPnP servers on the current network and allowed the user to pick a device (or a specific shared folder on that device) to be made available within the UPnP media source. Upon configuration, a node was added to the media source tree which directly connected to that folder allowing its contents to be played back by the media player.

Similarly, the Youtube configuration application (called “My YouTube”) allowed the user to generate a set of custom channels, each based on a custom search query. A “snowboarding” channel (with contents sorted by publication date) was created and used during the demo.

Since both the configurable media sources used for testing purposes supported pinnable media source nodes, numeric shortcuts were added to the custom nodes in order to make them look as linear channels.

As a final remark, both UPnP and YouTube channels were dynamic in nature, in that the contents they granted access to changed over time. In particular the UPnP channel, when selected from the media library, connected to the UPnP server to obtain the list of resources available at that particular time in the device/folder specified by the configuration application. Similarly, the search criteria associated with the custom YouTube channel were applied whenever the channel was invoked. In both cases, the multimedia contents were organized at runtime in a list of media elements and passed to the media player for continuous playback.

This inherent update capability of the custom channels was demonstrated during the demo by showing on the TV screen the most recent snowboarding video on YouTube, and the slideshow of the pictures taken during the presentation with a smart phone running a UPnP server service.

B. Beta release

The openBOXware framework has been released on March 1st, 2012 and can be installed on any Android device: <https://play.google.com/store/apps/details?id=it.uniurb.openboxware.launcher>. The Google Play marketplace has been populated with the core environment and some default add-ins (sample media sources and applications) that can be installed and integrated with the launcher. Stubs of representative use cases will be also published to provide a base for the development of other resources.

The current release is compatible with Android version 2.2.1 and superior (compatibility with Android ICS 4.0 has not been assessed yet) and it has been tested on a variety of devices, including: *Motorola Atrix* smartphone, based on nVidia Tegra 2 SoC (Dual ARM Cortex-A9, at 1 GHz), with 1 GB RAM, 4.0” screen (540×960), running Android 2.2; *Asus Transformer TF101* tablet/notepad, based on nVidia Tegra 2 SoC (Dual ARM Cortex-A9 at 1 GHz), with 1 GB RAM, 10.1” screen (1280×800), running Android

3.2; *Samsung Galaxy SII* smartphone, based on Exynos SoC (Dual ARM Cortex-A9 at 1.228 GHz), with 1 GB RAM, 4.3" screen (480×800), running Android 2.3; and an Android IP-box (<http://www.artwaytech.com/goodpro.php?id=266>) based on Samsung PV210 SoC (Cortex A8 CPU at 1 GHz), with 512 MB RAM, 2GB Flash memory, HDMI video & Audio output, running Android 2.2.

V. CONCLUSIONS

The advent of digital broadcast television, the diffusion of mobile broadband networks, the computational power of smartphones, and the success of open application frameworks have enabled the de facto convergence between mobile devices and set-top boxes and between broadcast television and online multimedia contents. Such a convergence has been exploited so far to create a thriving market of connected devices (including the so-called IP-boxes and smart-TVs) and to enhance the usage experience of television viewers (by making available additional IPTV and Internet TV channels, and by granting Internet browsing capabilities to any television set). Taking a different perspective, however, the same enabling conditions could be exploited to enhance usability and to reduce device diversity.

This is the starting point of this paper, further supported by two observations: first, watching television is a much more inclusive experience than browsing the Internet; second, in spite of the proliferation of any sort of connected devices, smartphones are the preferred ones with clear competitive advantages which prevent them to be outstripped in the near future.

This paper has introduced openBOXware for Android, an application suite that can be easily installed in any Android device (including a smartphone) to make it work as a set-top box while also maintaining compatibility with all the applications installed in it. The openBOXware core encompasses a launcher, designed to provide a lean-back usage experience in order to take advantage of the personalized runtime environment of the smartphone while watching television in the comfort of a living room, and a SDK that can be exploited to develop media sources and Android applications compatible with a TV-like usage mode.

The paper has outlined the key features of openBOXware, discussed the implementation choices, and presented representative use cases. In particular, it has been shown that openBOXware provides the opportunity of creating custom TV channels made of linearized contents possibly taken from heterogeneous sources (including local file systems, UPnP servers, streaming servers, and HTTP servers). Custom channels can be associated with numeric short cuts in order to make it possible to directly zap into them using a standard remote control. This makes it possible for elders and kids who are not used to browse the Internet to gain access to online contents organized in personalized linear TV channels by their family members.

OpenBOXware will be available on the Android market since March 1, 2012.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the EU IST Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n 25741, project ULOOP (User-centric Wireless Local Loop), and from the Italian ICT4University Programme, project U4U (University for University). The authors would like to thank RAI Radiotelevisione Italiana, BAX srl, and IMAB Group SpA for taking part in the setup of the demo, as well as Beatrice Bucciarelli and Lorenzo Bravi for their contribution to the implementation of openBOXware.

REFERENCES

- [1] L. Klopfenstein, S. Delpriori, G. Luchetti, E. Lattanzi, and A. Bogliolo, "Making an Android Tablet Work as a Set-Top Box," in *Proceedings of the International Conference on Advances in Future Internet*, ser. AFIN-2011. IARIA, 2011, pp. 64–68.
- [2] K. Mikkonen, "Exploring the creation of systemic value for the customer in advanced multi-play," *Telecommunications Policy*, vol. 35, no. 2, pp. 185 – 201, 2011.
- [3] L. Zhou, A. V. Vasilakos, L. T. Yang, and N. Xiong, "Multimedia Communications over Next Generation Wireless Networks," *EURASIP Journal on Wireless Communications and Networking*, 2010.
- [4] Akamai, "Q3 2010 - The State of the Internet," *Akamai report*, 2011.
- [5] Cisco, "Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015," *Cisco White Paper*, 2011.
- [6] A. Holzer and J. Ondrus, "Mobile Application Market: A Mobile Network Operators' Perspective," in *Exploring the Grand Challenges for Next Generation E-Business*, ser. Lecture Notes in Business Information Processing, W. Aalst *et al.*, Eds. Springer Berlin Heidelberg, 2011, vol. 52, pp. 186–191.
- [7] C. Maturana, A. Fernandez-Garcia, and L. Iribarne, "An implementation of a trading service for building open and interoperable dt component applications," in *Trends in Practical Applications of Agents and Multiagent Systems*, ser. Advances in Intelligent and Soft Computing, J. Corchado *et al.*, Eds., 2011, vol. 90, pp. 127–135.
- [8] A. Schroeder, "Introduction to MeeGo," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 4–7, 2011.
- [9] D. Gavalas and D. Economou, "Development platforms for mobile applications: Status and trends," *IEEE Software*, vol. 28, no. 1, pp. 77–86, 2011.
- [10] E. Tsekles, R. Whitham, K. Kondo, and A. Hill, "Investigating media use and the television user experience in the home," *Entertainment Computing*, 2011.
- [11] M. Butler, "Android: Changing the Mobile Landscape," *IEEE Pervasive Computing*, vol. 10, no. 1, pp. 4–7, 2011.

Enhanced QoS and QoE Support in UMTS Cellular Architectures Based on Application Requirements and Core Network Capabilities

An Autonomic Resource Management Perspective

Emanuel Puschita, Andra Elena Iulia Pastrav,
Cristian Androne, Tudor Palade

Communications Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

Emanuel.Puschita@com.utcluj.ro,
Andra.Pastrav@yahoo.com,

Cristian.androne@com.utcluj.ro, Palade@com.utcluj.ro

Alexandru Caruntu

Research & Development Department

Nokia Romania SRL

Cluj-Napoca, Romania

Alexandru.Caruntu@nokia.com

Abstract—The goal of this paper is to propose a solution in order to enhance the quality of services support inside the Internet Protocol-based packet switched domain of an Universal Mobile Telecommunications System architecture. The originality of this approach lies in the perspective of an integrated functionality defined at the conjunction of the service requests with the network capabilities. Objective quality of services and quality of experience analysis will highlight the benefits of the autonomic resource management in terms of average end-to-end delay, jitter and mean opinion score.

Keywords—UMTS; QoS support; QoE perception; autonomic management.

I. INTRODUCTION

The success of the Internet Protocol (IP) technology, as evidenced by the variety of many types of applications and network architectures, has proven its centrality through the way it has influenced, and often determined the daily life.

Continuous user demands for traffic capacity determined the operators and infrastructure providers to identify and to offer end-to-end solutions including carefully-managed connections and high-quality user services and experience. In this context, Internet's ubiquity has revealed a number of shortcomings that the current architecture cannot solve.

Hence a multitude of solutions have been developed to approach problems such as addressing, routing, congestion, resource management or traffic precedence.

In a sustained effort to improve resource management for mobile network environments, our previous work was focused on the benefits offered by an integrated Quality of Services (QoS) support inside the Universal Mobile Telecommunications System (UMTS) IP packet switched (PS) part of the Core Network (CN) domain [1]. Introducing a realistic radio access network modeling, the work presented in this paper completes over the quality of delivered services.

In this way, we demonstrate the need for a correlation between the application request and the network context as

the basic background of an integrated autonomic resource management.

A similar solution that considers the needs for built in quality control functions needed by many applications, addresses the resource management issue by combining a routing procedure and an information model for the representation of connection properties [2].

Two major trends can be distinguished in the way the scientific community understood solving this set of problems, namely: the complete remodeling of the Internet architecture (i.e., 4WARD [3], AUTOI [4], 4D [5], GENI [6]), and the gradual improvement of functionalities in the existing architecture [7] (i.e., Self-NET [8]).

The complete remodeling of the Internet architecture offers a purist approach, a clean slate kind of modeling the new architectural elements. On the other hand, the gradual development of network architecture, ruled by a pluralistic approach, considers that the leap towards a new Internet architecture is impossible independently of the existing technologies.

Promoting the functionality promised by a clean slate approach (flexibility, reliability, fault-tolerance, autonomy and manageability), the authors of this paper believe however that passing to an architecture that will integrate all these features is progressive, at least for two reasons: the perspective of operators and Internet service providers on radical changes in the network and the difficulty in testing, evaluating and validating the proposed new architectural elements.

Beyond the need for new legislative and normative agreements between Internet service providers and network operators, agreements required by fundamental architectural changes. Therefore, a major issue in the revolutionary innovation of the Internet architecture is the difficulty of assessing the new concepts in real experimental scenarios.

In [9], Peterson disputes the promotion of new architectural ideas, calling the scientific community to test the proposed solutions in the experimental "M-Lab" test-bed site [10], a validation site completely different from what means the evaluation by simulation or emulation.

Although the reality of testing on an experimental platform is undeniable, the risk is to focus the proposed solutions on a single extremely narrow issue.

Therefore, the authors of this paper believe that prior to a live testing phase there are several steps that must be completed by simulation and emulation, namely: monitoring and highlighting critical situations to identify network problems, testing the effect of local parametric adjustments on the whole system, development and gradual integration of scalable features in a new architecture. Therefore, stepping towards a revolutionary architecture is a matter of time; the new capabilities added to the existing architectural elements represent the prerequisites for success in this matter. Because of this, we believe that it is impossible to jump towards an architecture, which is independent of the existing technologies, the argument of this statement being found in the evolutionary pluralist concept.

Starting from these premises, which combine the requirements of a clean slate paradigm with current technological reality, the paper aims to investigate and test the benefits of integrating autonomic resource management capabilities into the UMTS CN architectural domain to enhance the QoS support.

In addition to analyzing QoS parameters such as average end-to-end delay, throughput or average jitter experienced by time-critical applications in the UMTS radio access network domain (UMTS RAN) [11], this paper enhances and extends the QoS / Quality of Experience (QoE) support even to the UMTS IP PS domain by integrating an autonomic resource management through network virtualization.

As the native UMTS QoS support is based only on service level classification in the Radio Access Network (RAN) domain, by describing, transmitting and correlating particular requirements of the source application with the context of IP PS domain, an optimal end-to-end performance could be offered through network virtualization in the CN.

In the context of this paper, by a native UMTS QoS support we consider a UMTS system offering QoS traffic differentiation by default. It is worth mentioning that the UMTS traffic classification covers only a certain part of the system and is closer to the physical connection (the RAN part), and therefore always has more stringent requirements in terms of QoS parameters.

Section II of the paper describes the radio access techniques, the architectural elements and the capabilities of the UMTS network domains that support various QoS traffic classes.

A holistic view of the quality of delivered services through the UMTS QoS support is completed by a QoE perspective. The QoE concept, the evaluation models and the parameters involved in the QoS objective analysis are presented in Section III.

Section IV presents the system model calibration for UMTS RAN in terms of Cumulative Distribution Function (CDF) of the Signal-to-Noise Interference level (SINR) for users in one cell that maximize the performances of the UMTS air interface. The UMTS RAN system model parameters that guarantee an optimal dimensioning of the network (transmission power, accepted co-channel

interference levels, radio range and cell capacity) are determined. Then, the performances of QoS/QoE support traffic classes for applications that have stringent requirements concerning the time component are evaluated.

The UMTS RAN dimensioning is performed by using Matlab, while the end-to-end system performance evaluation is performed by using QualNet 5.1 network simulator [12].

In Section V, the premises for an autonomic resource management that ensures a higher quality support in the UMTS CN are investigated in terms of QoS/QoE support.

This fact is accomplished by the ability of the application to select an alternative route between UMTS CN entities, the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN), based on its knowledge of the source requirements.

Finally, Section IV presents the conclusions of the conducted study by showing the perspective of an autonomic management of network resources that is based on the conjunction of application requirements and network context.

II. AN OVERVIEW OF THE UMTS CELLULAR SYSTEM

We will present the radio access techniques, the architectural elements and capabilities of the UMTS network domains that support various QoS traffic classes.

A. UMTS Radio Access Techniques

The Universal Mobile Telecommunications System is one of the third generation (3G) cellular system technologies. It uses Wideband Code Division Multiple Access (WCDMA), a direct-sequence spread spectrum access technology developed by NTT DoCoMo [13].

NTT DoCoMo submitted WCDMA specification to the International Telecommunication Union (ITU) as a proposal for the air interfaces of the ITU IMT-2000 family of 3G Standards. Therefore WCDMA was selected as an air interface for UMTS system, the 3G successor to Global System for Mobile Communications (GSM).

Since it is GSM down-compatible, some of the UMTS key features include the two basic multiplexing modes, Time Division Duplexing (TDD) and Frequency Division Duplexing (FDD). The advantages offered by the use of WCDMA, a time-frequency space multiplexing technique were reflected in extended network coverage and increased cell capacity. This is a direct consequence of the accepted cell interference vs. number of mobile users in the system compromise. The benefits of using WCDMA over other multiple access techniques also reflect on adaptive power control, variable transmission rates and inherent QoS support.

B. UMTS Network Architecture

The UMTS network architecture consists of three interacting parts: the User Equipment (UE) or Mobile Equipment (ME), the UMTS Terrestrial Radio Access Network (UTRAN) and the Core Network (CN). Due to its various radio access supported methods, the UMTS UE is capable of working in three modes: Circuit Switching (CS), Packet Switching (PS) and hybrid CS/PS mode [14].

The RAN part consists of the Base Stations (BSs) or Node Bs and Radio Network Controllers (RNCs). The main functions of the BS are air interface transmission / reception, signal modulation / demodulation, while major functions of the RNC are power control, admission control, channel allocation, radio resource control / management, data multiplexing / demultiplexing.

The CN part provides data switching, routing and handles the user traffic. It also contains the network user registries or databases and the network management functions. Offering both CS and PS, the CN is divided in two domains: CS domain and PS domain. The CS domain network elements are the Mobile Services Switching Centre (MSC), Visitor Location Register (VLR) and Gateway MSC (GMSC). The PS elements are the SGSN and the GGSN. Network operational and maintenance elements like HLR, VLR, Equipment Identifier (EIR) and Authentication Center (AuC) are shared by both domains.

Concluding, we have to note that the UMTS CN part is based on the GSM network architecture with General Packet Radio Service (GPRS) support.

In order to achieve a certain QoS support, UMTS network has defined a so-called "Bearer Service". A bearer service includes all aspects needed to enable the provision of a contracted QoS, including the control signaling, user plane transport and QoS management functionality.

C. UMTS QoS Support

Various types of bearer service were established between different parts of UMTS network. In addition, each bearer service on a specific layer offers its individual services using services provided by the layers below.

It is worth mentioning that bearers that cover only a certain part of a system and are closer to the physical connection always have more stringent QoS requirements.

In order to solve the QoS problem, UMTS defines four types of traffic classes: Conversational (CO) Class, Streaming (ST) Class, Interactive (IN) Class and Background (BK) Class [15]. The main difference between these QoS classes is the transfer delay value. Conversational QoS Class includes real-time applications that require stringent limits for delay value, while Background QoS Class is the most delay insensitive traffic class.

Table I shows the main characteristics of the above-mentioned QoS classes and examples of corresponding applications.

TABLE I. MAIN CHARACTERISTICS OF UMTS QoS CLASSES

Traffic classes	Characteristics	Application
Conversational (CO) Class	low delay, low jitter, symmetric traffic, no buffering	speech, voice over IP (VoIP), video, video gaming
Streaming (ST) Class	moderate delay, moderate jitter, asymmetric traffic, buffering allowed	multimedia, video streaming, audio streaming, video on demand
Interactive (IN) Class	moderate jitter, asymmetric traffic, buffering allowed, request response pattern	web browsing, network gaming, databases access

Background (BK) Class	destination doesn't expect data within a certain time, preserve payload content, asymmetric traffic, buffering allowed	email, file downloading, fax, short message services (SMS)
-----------------------	--	--

In order to define the traffic characteristics, the UMTS network architecture introduces a set of QoS attributes. It must be mentioned that a particular type of traffic class is itself a QoS attribute.

There are attributes specific to all classes (i.e., maximum bit rate, delivery order, maximum SDU (Service Data Unit) size) and some attributes that are applied only to a specific class (i.e., transfer delay is applied only to conversational and streaming classes; traffic handling priority is applied only to interactive class).

III. QUALITY OF EXPERIENCE

QoE is defined by the ITU-T as "the overall acceptability of an application or service, as perceived subjectively by the end-user" [16], thus it measures the performance expectations of the user. It is also stated that QoE includes "the complete end-to-end system effects", meaning that the source quality, the effects of the network, protocols, source codecs, terminals equipments etc. are reflected in the quality perception at the end-user.

Although it may seem that it overlaps with the QoS notion, QoE is not limited to the performances of the network. The end-user perception is also influenced by non-technical aspects such as environmental, sociological, and psychological factors, and thus it would be more accurate to say that QoE augments QoS by linking the performance of the system and the user's expectations.

As expected, due to the fact that it reflects the user's opinion regarding the quality of the transmission, QoE cannot be easily evaluated. However, two types of evaluation methods are usually approached: the subjective and the objective one.

In order to evaluate the quality of a transmission from a subjective perspective, different tests and experiments have to be conducted using human subjects. This tests directly ask the participants to rate their experience regarding a certain service. Although this method is the only way of assessing the psychological and sociological impacts on QoE, it is expensive and time consuming [17]. Consequently, the second evaluation method is used in order to implement several models that associate the network performance with the user's level of satisfaction.

As described in [18], in the process of developing such a model, three main steps have to be considered: (1) analyze key QoS parameters that have an impact on the performance perceived by the user (e.g. delay, jitter or packet loss [19]) and identify the relationship between them and QoE, (2) measure the parameters considered in the first step and (3) use mapping metrics to rate the QoE based on the QoS measured parameters.

A. Objective QoE Analysis: The E-model

The most popular model used to predict the quality of a voice application is the E-model [20]. It is an objective

method used to compute the performances of an end-to-end voice transmission and thus anticipate the quality perceived by the end-user taking into account the network impairment parameters like packet loss and delay.

The primary outcome of the E-model is the Rating Factor R as in (1)

$$R = R_0 - I_s - I_d - I_{e-eff} + A, \quad (1)$$

where R_0 is the Signal to Noise Ratio, I_s represents the impairments that occur at the same time as the voice signal, I_d represents the impairments caused by the delay, I_{e-eff} is the effective equipment impairment factor (caused by the low rate codecs), and A is the advantage factor, which may compensate for some of the impairment factors assuming that there are other advantages of access to the user [21].

By processing the R factor, an estimation of the user's opinion can be obtained.

B. Mean Opinion Score

The Mean Opinion Score (MOS) is the most used QoE metric and is defined in [16] as “the value on a predefined scale that a subject assigns to his opinion of the performance of the telephone transmission system used either for conversation or only for listening to spoken material”.

Although this definition refers to the telephone transmission system, the MOS is used in the evaluation of voice and video applications too. MOS is expressed as a number between 1 and 5, from the lowest to the highest perceived quality.

Table II presents the correspondence between the absolute value of MOS, the perceived quality descriptor and the degradation of the transmission as perceived by the user.

TABLE II. MOS CORRESPONDENCE TABLE

MOS	Quality descriptor	Degradation
5	Excellent	Imperceptible
4	Good	Perceptible
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

A correspondence between the value of the R factor and MOS [20] is given in (2):

$$\begin{aligned} R < 0, \text{ MOS} &= 1, \\ 0 < R < 100, \text{ MOS} &= 1 + 0.035R + R(R - 60)(100 - R)7 \cdot 10^{-6}, \quad (2) \\ R > 100, \text{ MOS} &= 4.5. \end{aligned}$$

C. QualNet 5.1 implementation of MOS

In addition to describing the subjective opinion, MOS is also used for scores that originate from objective models, like the one implemented by Scalable Networks for QualNet 5.1 simulator.

QualNet Simulator provides several parameters describing the performances of the network in the simulated scenarios. Among the usual objective parameters, MOS is also computed, based on the ITU-T E-model presented earlier.

The R factor is computed in a simplified manner as in (3)

$$R = 93.2 - I_d - I_{ef}, \quad (3)$$

where I_d represents the impairment factor due to the delay in the network expressed by (4)

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3). \quad (4)$$

Parameter d is the one-way delay (including coding, network and de-jitter delay) and I_{ef} (5) is the impairment caused by the low bit rate

$$I_{ef} = 30 \ln(1 + 15e)H(0.04 - e) + 19 \ln(1 + 70e)H(e - 0.04), \quad (5)$$

where e is the error probability and $H(x) = 0$ for $x < 0$ and $H(x) = 1$ for $x \geq 0$ [12]. The formulae for computing MOS by means of the R factor are presented in (2).

IV. UMTS RAN SYSTEM MODEL AND NATIVE UMTS QoS SUPPORT EVALUATION

In this section, the system model calibration for UMTS RAN in terms of CDF of the SINR level for users in one cell that maximize the performances of the UMTS air interface is determined. Then, the performances of QoS/QoE support traffic classes for applications that have stringent requirements concerning the time component are evaluated.

A. Radio Access System Model Parameter Calibration

The air interface calibration scenario considered in this paper consists of a $1.4 \times 1.4 \text{ km}^2$ area, in which a UMTS RAN has been deployed.

The UMTS RAN calibration system model considers seven Node Bs deployed, each of them having three-sector antennas. The Inter-Site Distance (ISD) is considered to be 500m, as for a typical urban environment. Furthermore, the NodeBs transmit powers are set to 20W.

The radio propagation model considered in the simulations consists of a log-distance path loss model with fixed and exponential coefficients taken from [21].

The amplitude change caused by shadowing is modelled using a log-normal distribution with a standard deviation according to the log-distance model. The shadow fading maps were generated based on the method in [22] with the standard deviation parameters taken from [21]. The summary of the full range of parameters used in the simulations is presented in Table III.

TABLE III. UMTS RAN SIMULATION PARAMETERS

Path loss	Path-loss is modelled as $11.81 + 38.63 \log_{10}(d)$ for network users, where d is the distance from the base station in meters.
Shadow fading	Shadow fading is modelled as spatially correlated random process with log-normal distribution (6dB standard deviation), spatial correlation $r(x) = e^{-\sqrt{20}}$ for distance x .
Receiver noise power	The receiver noise power is modelled as $10 \log_{10}(kT NF W)$ where the effective noise bandwidth is given as $W = 3.84 \times 10^6$ Hz, and $kT = 1.3804 \times 10^{-23} \times 290$ W/Hz. The noise figure at the UE is $NF_{[dB]} = 7$ dB.
Base station antenna gain	The base station antenna gain is calculated as $G(\theta)_{dB} = G_{\max} - \min \left[12 \left(\frac{\theta}{\beta} \right)^2, G_s \right], \quad -\pi \leq \theta \leq \pi \quad \text{with}$ $\beta = 70\pi / 180 \quad \text{angle where gain pattern is 3dB down from peak}$ $G_s = 20 \text{ dB} \quad \text{sidelobe gain level in dB}$ $G_{\max} = 16 \text{ dB} \quad \text{maximum gain level in dB}$

Radio system simulations were carried out in order to investigate the performances of the UMTS mobile network. The simulation parameters were taken from Table III, and if not otherwise mentioned they will be the same for all the simulations carried out in the paper.

The received signal strength on the downlink direction is calculated as in (6)

$$P_{Rx} = P_{tx} + P_{loss} + P_{shadow} + P_{antenna}, \quad (6)$$

where P_{tx} is the transmit power of the base station; P_{loss} is the path loss component calculated according to Table III; P_{shadow} is the shadow fading component at the user location defined by the parameters in Table III; $P_{antenna}$ is the antenna gain component specific for the three sector antenna used at the Node B transmitter.

In order to have a general view of the system model, Figure 1 presents the level of the received signal strength throughout the environment. The levels for the received signal are calculated according to equation (6). For a better understanding and clarity of the figure, the values of the signal displayed in Figure 1 are calculated without considering any shadow fading.

The values of the received signal strength are given in dBm, according to the colour bar on the right hand side of the figure. The resolution of the map is 2m.

Figure 1 presents the general view of the system model, but in order to understand the performances of the network we need to analyse it cell-wise. In order to do this, we will further concentrate on one of the three hexagonal cells of the centre base station.

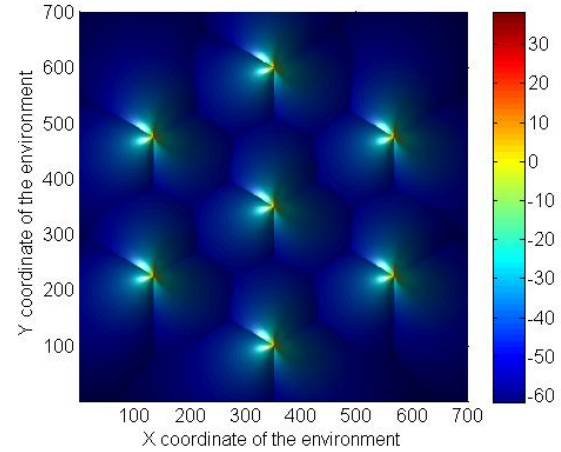


Figure 1. Received signal strength level for the considered environment

The analyzed parameter in this case will be the signal to interference plus noise ratio (SINR), defined as in (7)

$$SINR = P_{Rx_max} - (P_{interference} + P_{noise}), \quad (7)$$

where P_{Rx_max} is the received signal strength from the best server; $P_{interference}$ is the sum of the contributions of all the other base stations; P_{noise} is the receiver noise power defined according to Table III.

Using (6), the SINR level will be calculated for all the positions in the environment with the same location resolution. Because the level of the noise power at the user location is -101.13dBm, as calculated from the relations in Table III, the receiver sensitivity considered in the scenario needs to be above this threshold.

One factor, which has a great impact upon the obtained SINR value, is the shadow fading. This phenomenon occurs when an obstacle is situated between the transmitter and the receiver, in our case base station and mobile terminal, respectively. In order to better understand the influence of the shadow fading component upon the level of the obtained SINR, we will vary the value of the shadow fading standard deviation. The analysis we conducted led us to the results presented in Figure 3, which illustrates the CDF of the obtained SINR value for all the user positions in one of the hexagonal centre cells.

The results are calculated for the instances when we have no shadow fading, shadow fading with standard deviation 4, 6, 8 and 10, respectively.

The analysis reveals that the shadow fading has a negative impact upon the obtained SINR value, directly proportional to the increase of the standard deviation.

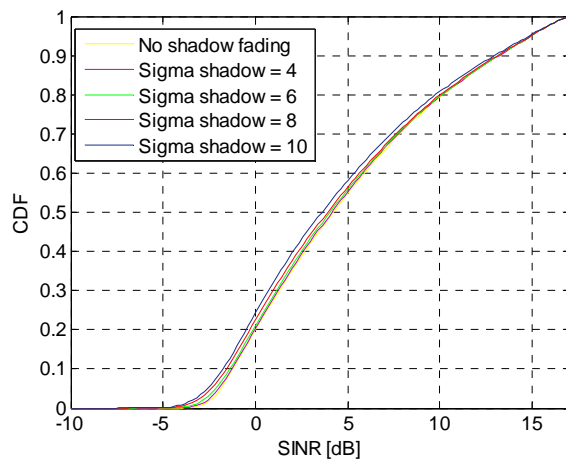


Figure 2. CDF of SINR for users in one cell

Because in our scenario we consider a minimum threshold value of the SINR of -6.5dB, the worst case that fulfils this condition is that of using a standard deviation of 6. The minimum value of -6.5dB for the SINR, corresponds to a user throughput of 500kbps.

B. Performance Evaluation of UMTS QoS/QoE Support

In order to evaluate the QoS support implicitly provided by a UMTS network, a scenario including a PLMN (Public Land Mobile Network) was simulated using QualNet 5.1 network simulator, a widely used platform in the defense and telecommunication network design and evaluation [12].

Global parameters configured at the physical UMTS RAN level of the simulation are given in Table IV.

TABLE IV. PHY LAYER CONFIGURATION PARAMETERS

Parameter	Value
Terrain dimensions	1.4 x 1.4 km ²
Up-link channel frequency	1.95 GHz
Down-link channel frequency	2.15 GHz
Shadowing Mean	6 dB
Node B transmission power	20 W
Simulation time	200 s

The system model parameters previously determined guarantee an optimal dimensioning of the network in terms of signal distribution vs. system capacity. It should be noticed that two different radio channels were used in order to access the network resources. The frequency values of these channels were chosen accordingly with European 3G bands for UMTS 2100 recommendations [23].

The UMTS RAN network scenario includes eight UE nodes: four source nodes (UE nodes 6, 8, 10, and 12) and four destination nodes (UE nodes 7, 9, 11, and 13), as presented in Figure 3.

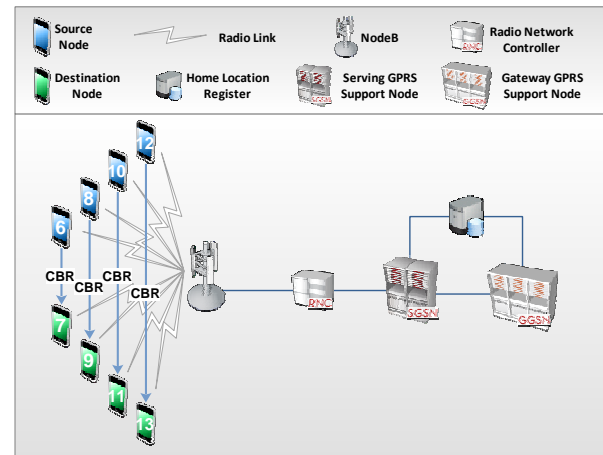


Figure 3. UMTS evaluation scenario

Between the source nodes (SN) and destination nodes (DN), four CBR (Constant Bit Rate) applications, each of them corresponding to one QoS class defined by the UMTS network, were considered. The characteristics of all applications are summarized in Table V.

TABLE V. CORRESPONDING QoS CLASS FOR EACH CBR APPLICATION USED IN THE EVALUATION SCENARIO

Application Type	SN	DN	Items to Send	Item Size (bytes)	Interval (s)	QoS Class
CBR	6	7	1000	40	0.1	BK
	8	9				IN
	10	11				ST
	12	13				CO

The results of the simulations concerning the average end-to-end delay, the average jitter and average MOS for each QoS supported class are presented in Figure 4, Figure 5 and Figure 6 respectively.

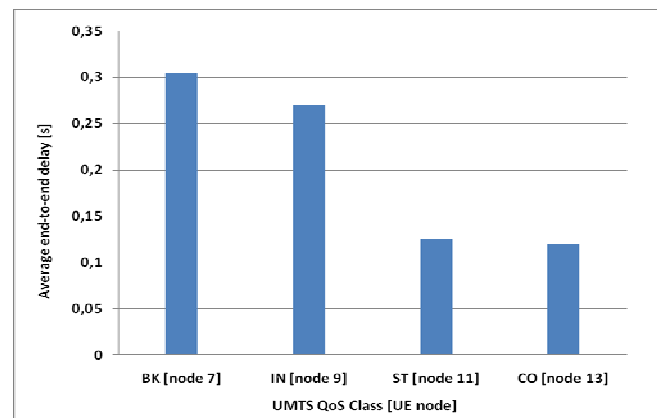


Figure 4. Average end-to-end delay experienced by the test applications in each QoS support class

In analyzing the obtained results, it can be noticed that application, which corresponds to the Conversational and

Streaming QoS class, is characterized by the lowest value of average end-to-end delay and jitter delay, as expected for the type of application corresponding to this class (speech, VoIP, video or audio streaming).

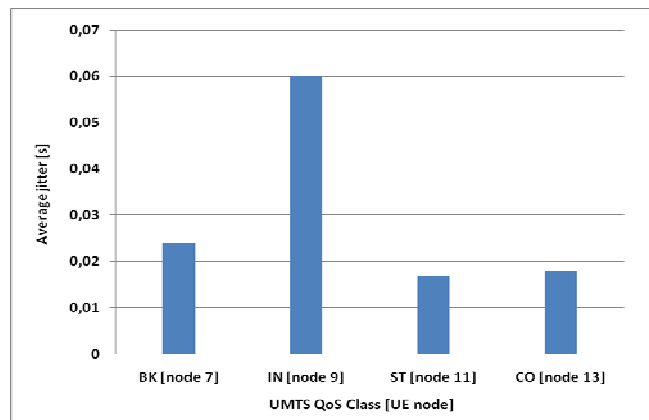


Figure 5. Average jitter experienced by the test applications in each QoS support class

Summarizing, the network complies with the priority level imposed on the test applications that were modeled by CBR-type traffic sources, and this is reflected in the average values of the end-to-end transmission delays and in the jitter for each QoS priority class.

Correlating end-to-end transmission delays with the transmission flow rate and with the time interval between transmitted packets one can notice the increased flow rate in the case of Background class, which is due to transmitting a reduced total number of packets in a very short transmission time interval.

If MOS values are compared, it can be seen that the application from Conversational QoS class has the highest value of this parameter, which corresponds to the lowest value for the average end-to-end delay and the average jitter.

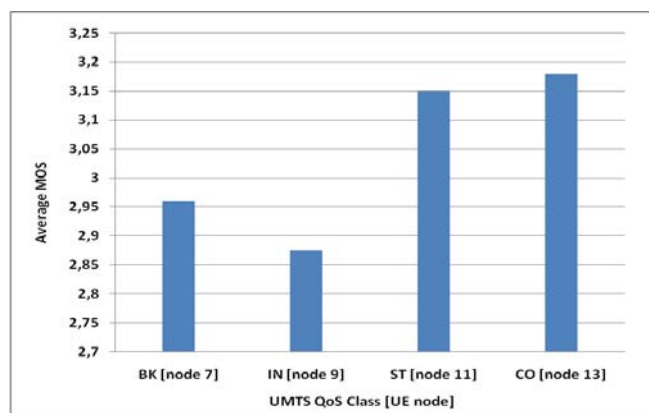


Figure 6. Average MOS experienced by the test applications in each QoS support class

Note that the QualNet simulator offers the possibility to mark the traffic flows by setting the appropriate IP

Precedence field. In this way, the applications are identified, classified and scheduled in the corresponding queues.

V. PERSPECTIVES TOWARDS AN AUTONOMIC RESOURCE MANAGEMENT IN A UMTS NETWORK

Developed within the 3rd Generation Partnership Project (3GPP), the 3G standard suggests an end-to-end QoS support based on a policy management system (Policy-Based Network Management) [15].

The network architecture presented by the first 3GPP public versions has evolved to an architectural model System Architecture Evolution (SAE) [24], which ensures the convergence of different access network categories such as UMTS, 3GPP, Wireless Local Area Network (WLAN) or any other non-3GPP radio access technology.

The latest public 3GPP version considers the IP Multimedia Subsystem (IMS) [23] network architecture to be completely separated from the access technology, having the specific access functions isolated from the core network. By eliminating the hierarchical dependence between the Serving RNC (SRNC) and the corresponding SGSN, a more efficient resource allocation was offered.

In order to manage QoS resources, the SAE architecture integrates an informational QoS Information Function (QIF) function that interacts with all individual network models included within the IMS platform. This QoS resource management solution is an external part of the network, based on the use and interaction between a central entity and periferic elements.

Although in the clean slate approach the resource management and the QoS support are considered an integrated part of communications networks, this fact is not reflected in the characteristics of current systems or by the UMTS network architecture.

Therefore, the perspective of an autonomic resource management in a UMTS network proposed in this paper suggests the necessity of adding additional information at the level of the central UMTS network elements using virtualization technique.

Network virtualization represents a high level abstraction process that overlaps the implementation and physical network configuration details. Allowing co-existence of multiple virtual architectures overlaid on a common substrate physically shared, network virtualization promises flexibility and security, promoting diversity and increased management capacity [24].

In this way, UMTS core network nodes act autonomously, being able to sense the environment, to perceive the changes, to understand internal changes and to react in an intelligent manner by selecting the optimal path according to application requirements.

To demonstrate this, native UMTS QoS support is analyzed in comparison to the potential of the autonomic management offered through network virtualization, using QualNet network simulator [12]. Thus, in the case of an autonomic management system, the proposed analysis scenarios highlight the ability of selecting the best route according to the source application constraints in terms of maximum acceptable end-to-end delay.

A. Scenario description

According to [13], it is possible for an UMTS network to have multiple SGSNs and GGSNs entities, which can be co-located or can be interconnected via an IP subnetwork in order to increase the geographical area served by an operator.

Considering the second approach, an evaluation scenario was developed, in which the SGSN and the GGSN are interconnected via a simple IP sub-network that consists of seven generic routers denoted R1 to R7, as depicted in Figure 7.

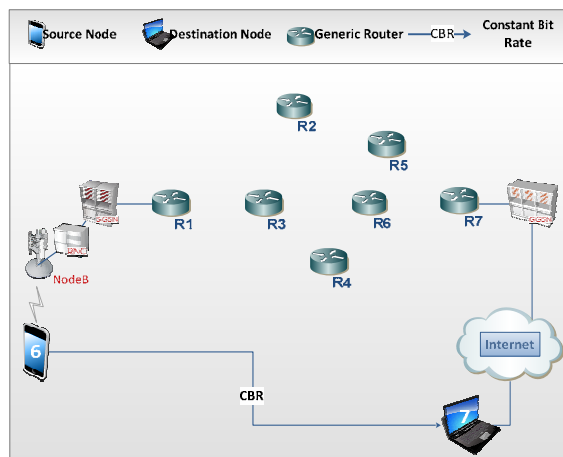


Figure 7. The architecture of the evaluation scenario

Obviously, a real-life UMTS IP sub-network would be more complex, the motivation for this topology was to better illustrate the problems that may arise and the proposed solution.

In this evaluation scenario a CBR test application corresponding to Conversational QoS class (highest priority QoS class) was considered. The main parameters that describe the characteristics for the tested application are indicated in Table VI.

TABLE VI. CHARACTERISTICS OF THE MODELED APPLICATION

Application Type	SN	DN	Items to Send	Item Size [bytes]	Interval [s]	QoS Class
CBR	6	7	1000	40	0.1	CO

As we have already mentioned, the motivation behind this evaluation is to highlight the benefits of the autonomic management offered through a network virtualization process.

In our case, the virtualization process could be illustrated by controlling, through the configuration files, the parametric values that characterize network architectural elements.

Knowing the values of these parameters, it is possible to indicate a dedicated virtual network that offers the best path from source to destination in terms of minimum average end-to-end delay, maximum throughput, packet loss rate on selected route or other stringent requirements specific to a certain type of application.

B. Performance Evaluation

In order to emphasize the path selection mechanism between SGSN and GGSN, core nodes of UMTS IP sub-network, two different cases were evaluated: a native UMTS QoS support selection compared to QoS network support provided by an autonomic network management selection, as a result of dedicated virtual network generation.

When it defines the links between two intermediate generic routers in the UMTS IP sub-network, the simulator allows the association of specific transmission throughput and delay on each link, as presented in Figure 8.

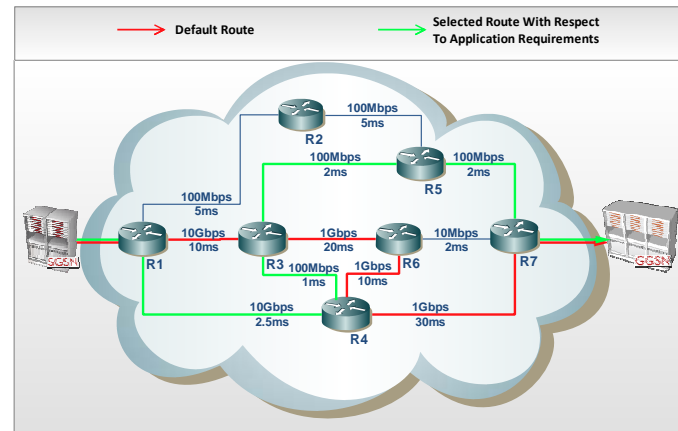


Figure 8. UMTS IP sub-network path selection

Sending the test application scheduled in Conversational Class by the native UMTS QoS support in the RAN domain, the path from router R1 to router R7 in the IP PS sub-network was selected via routers R3, R6 and R4, as illustrated in Figure 7. The path selection was based only on the use of OSPFv2 routing protocol. However, independently of the used routing protocol (e.g., Bellman Ford or RIP), simulation results indicates that there is no correlation between the application requirements and the network selected path. Therefore, the determined path remains the same in case of using default QoS support.

An autonomic network management QoS support should consider and accommodate both differentiated application requirements and dynamic network context. This mandatory integrated capability of the Future Internet (FI) network elements is implemented by means of network virtualization.

The objective of network virtualization process is to generate virtual networks and make each of these virtual networks appear to the user as a dedicated network infrastructure, with dedicated resources and services available for application requests. Therefore, the network virtualization process invokes a mode of selecting the virtual network that best integrates and satisfies the application requests at the physical network level.

It must be mentioned that for the generated virtual network, in our case, only the values of average end-to-end delay and jitter (as a consecutive delay difference) are considered as critical parameters for the source application.

Results of the simulations validate a virtual path from the router R1 to router R7 via routers R4, R3, and R5, a corresponding physical network infrastructure that offers best performances in terms of requested average end-to-end delay and jitter.

Parametric results of the average end-to-end delay, the average jitter, and the average MOS, both for native UMTS QoS support and autonomic resource management QoS support in the IP PS domain, are summarized in Table VII.

TABLE VII. PARAMETRIC EVALUATION OF QoS/QoE SUPPORT

Selected path between ingress-egress nodes in UMTS PS domain	Average end-to-end delay (s)	Average jitter (s)	Average MOS (score)
<i>Native QoS support in the UMTS radio access network</i>			
R1 → R3 → → R6 → R4 → R7	0.109	0.022	3.186
<i>Autonomic resource management based on network virtualization in UMTS IP PS domain</i>			
R1 → R4 → → R3 → R5 → R7	0.045	0.022	3.264

As illustrated in Figure 9 and Figure 10, the UMTS IP PS domain decisively influences the applications performances in terms of average end-to-end delay, jitter and MOS.

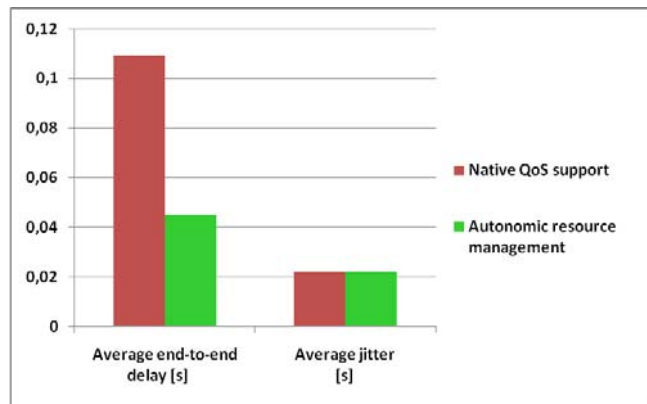


Figure 9. Comparative results of the average end-to-end delay and average jitter based on different path selection between ingress-egress nodes in the UMTS IP PS domain

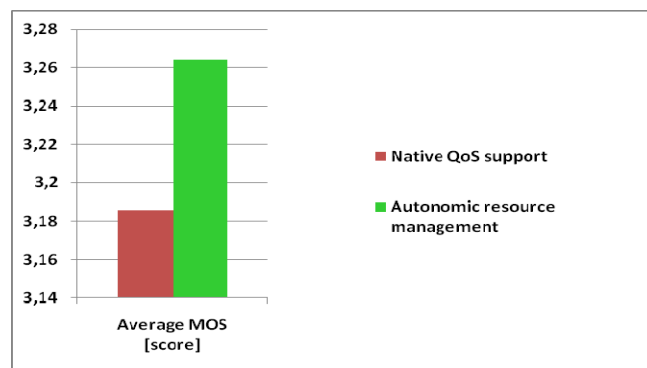


Figure 10. Comparative results of the average MOS based on different path selection between ingress-egress nodes in the UMTS IP PS domain

If the application requests are expressed in terms of a maximum accepted delay [26], the results illustrated in Figure 11 show that the autonomic resource management could satisfy these requests by offering an optimal path inside the UMTS IP PS domain.

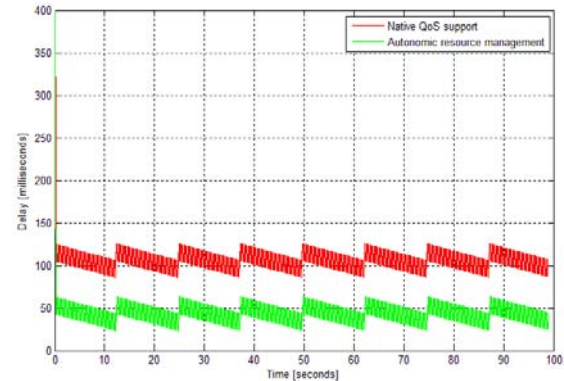


Figure 11. Delay variation over the simulation time based on different path selection between ingress-egress nodes in the UMTS IP PS domain

The QoE objective analysis based on MOS evaluation presented in Figure 12 shows that the user perception of the quality of the delivered service is enhanced by the use of an autonomic resource management support in the UMTS IP PS domain.

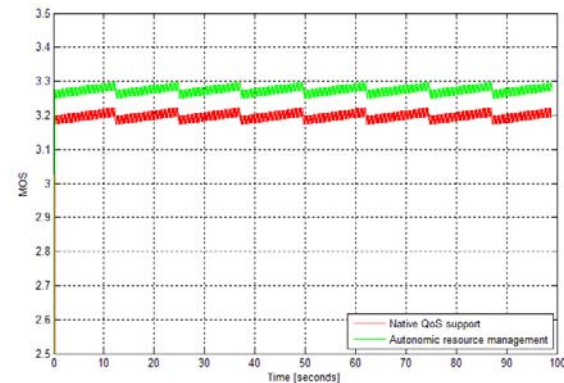


Figure 12. MOS variation over the simulation time based on different path selection between ingress-egress nodes in the UMTS IP PS domain

Therefore, simulation results support the perspective towards an autonomic resource management QoS support that makes the conjunction between the application's requests and the network context by choosing an alternate route in a virtualized environment.

The potential of the autonomic resource management is reflected by the capacity to identify an optimal path between the source node and the destination node, according to the imposed QoS constraints (for example, maximum accepted delay, minimum accepted packet loss, maximum accepted jitter or minimum requested throughput).

Some critical conditions or cost constraints could limit the resource optimization while using the native QoS support on the UMTS radio access network.

These situations request the adaptation of the source application parameters to the network available resources in order to still broadcast the information on the channel.

An implemented and tested solution (i.e., virtualization algorithm, virtualized parameters, in-network message flow, and source adaptation) for the network virtualization concept is offered in [27]. The conjunction between the source application requests and the network context is achieved through the QoS profiles message exchange and it represents the process of network virtualization.

As in [27], the paper assumes the network overflow by probing each UMTS IP PS network link. Nevertheless, the potential of the autonomic resource management is reflected by lower average end-to-end delay and lower jitter.

In order to impose the selected network elements on the virtual path, we used the Multiprotocol Label Switching (MPLS). As the proposed QoS support uses MPLS for route maintenance, it can be applied only in the UMTS IP PS network domain. Therefore, it cannot offer an end-to-end QoS support because RAN domain of the UMTS network does not have implemented the IP protocol.

VI. CONCLUSIONS AND FUTURE WORK

Considering a realistic UMTS RAN domain modeling, the paper aims to investigate the potential benefits that could reside from the integration of the autonomic resource management based on the virtualization process in current UMTS IP PS network domain architecture.

Based on an objective QoS/QoE performance analysis, the upper limit of native UMTS QoS support was analyzed, in a first stage, in terms of average end-to-end delay, average jitter and average MOS. The ability of the UMTS traffic classes to offer quality support for constant bit rate time critical applications was compared with the QoS support resulted from the usage of autonomic resource management.

The obtained results suggest that, since it is closer to the application needs and considering the network context, such an autonomic resource management could improve the native UMTS QoS support. Thus, the enhanced QoS/QoE support would overcome the situations in which the existent QoS mechanism would not even accept the service itself.

As a part of the further work, the authors intend to extend this solution also for the radio access part of the cellular networks. Moreover, future investigation will validate the simulation results through emulation on an experimental test-bed. This investigation will use the EXata emulation server (running the QualNet scenarios) and two operational hosts corresponding to the source and to the destination nodes (transmitting real-time traffic). In this way, a realistic experience of handling live traffic flows will be offered.

ACKNOWLEDGMENTS

This paper was supported by the following projects: "Development and support of multidisciplinary postdoctoral programmes in major technical areas of national strategy of Research - Development - Innovation" 4D-POSTDOC,

contract no. POSDRU/89/1.5/S/52603, project co-funded by the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013 and "Doctoral studies in engineering sciences for developing the knowledge based society-SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013. The logistics costs of the work (research infrastructure, conference fee and accommodation costs) were supported by CNCSIS-UEFISCSU, project number 184/2010.

REFERENCES

- [1] E. Puschita, G. Manuliac, T. Palade, and A. Caruntu, "QoS Support in UMTS Networks: Performance Evaluation and Perspectives towards an Autonomic Resource Management", The Third International Conference on Advances in Future Internet (AFIN 2011), ISBN: 978-1-61208-148-9, 2011, pp. 25-30.
- [2] M. Yampolskiy, W. Hommel, V. A. Danciu, M. G. Metzker, and M. K. Hamm, "Management-aware Inter-Domain Routing for End-to-End Quality of Service", International Journal on Advances in Internet Technology, vol 4, no. 1 & 2, ISSN: 1942-2652, pp. 60-78, 2011. http://www.iariajournals.org/internet_technology/index.html
- [3] C. Mingardi, G. Nunzi, D. Dudkowski, and M. Brunner, "Event Handling in Clean-Slate Future Internet Management," 11th IFIP/IEEE International Symposium on Integrated Network Management (IM 2009), June 2009, pp. 275-278.
- [4] L. Mamatas, S. Clayman, M. Charalambides, A. Galis, and G. Pavlou "Towards an Information Management Overlay for Emerging Networks," 12th IEEE/IFIP Network Operations and Management Symposium (NOMS 2010), April 2010, pp. 527-534.
- [5] A. Greenberg, et al., "A clean slate 4D approach to network control and management," ACM SIGCOMM Computer Communication Review, vol. 35, no. 5, October 2005, pp. 41-54.
- [6] D.H.C. Du, "Clean Slate Design of Internet for Supporting Service-Oriented Computing and Applications," IEEE International Conference on Service-Oriented Computing and Applications (SOCA 07), June 2007, pp. 3-8.
- [7] C. Dovrolis, "What would Darwin think about clean-slate architectures?," ACM SIGCOMM Computer Communication Review, vol. 38, no. 1, January 2008, pp. 29-34.
- [8] A. Kousaridas, et al., "Future Internet Elements: Cognition and Self-Management Design Issues", In ICST (The Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (ed.), Proceedings of the SAC-FIRE Workshop, Autonomics 2008 (sponsored by the European Commission), Turin, Italy, September 23-25, 2008.
- [9] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the Internet Impasse through Virtualization", Computer Journal, vol. 38, no. 4, 2005, pp. 34-41.
- [10] PlanetLab platform, <http://www.planet-lab.org/>, March 2011.
- [11] D. Kaur, P. Rakheja, and A. Kaur, "Analysing Quality of Service in UMTS", International Journal of Computer Applications, vol. 11, no. 12, 2010, pp. 31-34.
- [12] QualNet Developer 5.1 Network Simulator, <http://www.scalable-networks.com/>, March 2011.
- [13] C. Kappeler, UMTS network and beyond, John Wiley & Sons, West Sussex, 2009.

- [14] 3GPP TS 23.101, "General Universal Mobile Telecommunications System (UMTS) architecture", Release 10, 2010.
- [15] 3GPP TS 23.207, "Technical Specification Group Services and System Aspects; End-to-end Quality of Service (QoS) concept and architecture, Release 9, December 2009.
- [16] ITU-T Recommendation P.10/G.100, "Vocabulary for performance and quality of service", 2006.
- [17] R. Stankiewicz, P. Cholda, and A. Jajszczyk, "QoX: What is it really?", IEEE Communications Magazine, vol. 49, no. 4, April 2011, pp. 148-158.
- [18] D. Soldani, M. Li, and R. Cuny, (editors), "QoS and QoE Management in UMTS Cellular Systems", John Wiley & Sons, 2006.
- [19] ITU-T Recommendation G.1010, "End-user multimedia QoS categories", 2001.
- [20] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning", 2005.
- [21] 3GPP TR 36.814, "Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects", Release 9, 2010.
- [22] H. Claussen, "Efficient modelling of channel maps with correlated shadow fading in mobile radio systems", in Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2005.
- [23] 3GPP TS 23.228, "IP Multimedia Subsystem (IMS); Stage 2", Release 10, September 2010.
- [24] 3GPP TR 23.882, "3GPP system architecture evolution (SAE): Report on technical options and conclusions", Release 8, September 2008.
- [25] A. G. Prieto, et al., "Decentralized In-Network Management for the Future Internet", IEEE International Conference on Communications (ICC'09), 2009, pp. 1-5.
- [26] N. Jingjing Zhang Ansari, "On assuring end-to-end QoE in next generation networks: challenges and a possible solution", IEEE Communications Magazine, vol. 49, no. 7, 2011.
- [27] E. Puschita, T. Palade, A. Moldovan, R. Colda, and I. Vermesan, "An Innovative QoS Paradigm based on Cognitive In-Network Management of Resources for a Future Unified Network Architecture: I-NAME QoS Model", The Second International Conference on Advances in Future Internet (AFIN 2010), ISBN 978-0-7695-4091-7, 2010, pp. 37-43.



www.iariajournals.org

International Journal On Advances in Intelligent Systems

✦ ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, ENERGY, COLLA, IMMM, INTELLI, SMART, DATA ANALYTICS

✦ issn: 1942-2679

International Journal On Advances in Internet Technology

✦ ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING, MOBILITY, WEB

✦ issn: 1942-2652

International Journal On Advances in Life Sciences

✦ eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO, SOTICS, GLOBAL HEALTH

✦ issn: 1942-2660

International Journal On Advances in Networks and Services

✦ ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION, VEHICULAR, INNOV

✦ issn: 1942-2644

International Journal On Advances in Security

✦ ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

✦ issn: 1942-2636

International Journal On Advances in Software

✦ ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, IMMM, MOBILITY, VEHICULAR, DATA ANALYTICS

✦ issn: 1942-2628

International Journal On Advances in Systems and Measurements

✦ ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL, INFOCOMP

✦ issn: 1942-261x

International Journal On Advances in Telecommunications

✦ AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA, COCORA, PESARO, INNOV

✦ issn: 1942-2601