EMERGING 2014

The Sixth International Conference on Emerging Network Intelligence

ISBN: 978-1-61208-357-5

August 24 - 28, 2014

Rome, Italy

EMERGING 2014 Editors

Eric Veith, Wilhelm Büchner Hochschule, Germany

Pascal Lorenz, University of Haute Alsace, France

EMERGING 2014

Forward

The Sixth International Conference on Emerging Network Intelligence (EMERGING 2014) held on August 24 - 28, 2014 - Rome, Italy, constituted a stage to present and evaluate the advances in emerging solutions for next-generation architectures, devices, and communications protocols. Particular focus was aimed at optimization, quality, discovery, protection, and user profile requirements supported by special approaches such as network coding, configurable protocols, context-aware optimization, ambient systems, anomaly discovery, and adaptive mechanisms.

Next-generation large distributed networks and systems require substantial reconsideration of existing ‘de facto’ approaches and mechanisms to sustain an increasing demand on speed, scale, bandwidth, topology and flow changes, user complex behavior, security threats, and service and user ubiquity. As a result, growing research and industrial forces are focusing on new approaches for advanced communications considering new devices and protocols, advanced discovery mechanisms, and programmability techniques to express, measure, and control the service quality, security, environmental and user requirements.

We take here the opportunity to warmly thank all the members of the EMERGING 2014 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the EMERGING 2014. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the EMERGING 2014 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the EMERGING 2014 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in emerging technologies.

We hope Rome provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

EMERGING 2014 Chairs

EMERGING Advisory Chairs

Michael D. Logothetis, University of Patras, Greece
Tulin Atmaca, IT/Telecom&Management SudParis, France
Carl James Debono, University of Malta, Malta
Robert Bestak, Czech Technical University in Prague, Czech Republic
Zoubir Mammeri, IRIT - Toulouse, France
Raj Jain, Washington University in St. Louis, USA
Phuoc Tran-Gia, University of Wuerzburg, Germany
Norihiko Yoshida, Saitama University, Japan
António Nogueira, DETI-University of Aveiro/Instituto de Telecomunicações, Portugal
Ioannis Moscholios, University of Peloponnese, Greece
Henrik Karstoft, Aarhus University, Denmark
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Anne James, Coventry University, UK
Anna Medve, University of Pannonia, Hungary
Nikolaos Tselikas, University of Peloponnese, Greece
Jelena Zdravkovic, Stockholm University, Sweden
Rolf Drechsler, University of Bremen/DFKI, Germany
Christian Blum, IKERBASQUE - Basque Foundation for Science University of the Basque Country, Spain

EMERGING Industry/Research Chairs

Robert Foster, Edgemount Solutions - Plano, USA
David Carrera, Barcelona Supercomputing Center (BSC) / Universitat Politecnica de Catalunya (UPC), Spain
Preetha Thulasiraman, Naval Postgraduate School - Monterey, USA
Anastasiya Yurchyshyna, University of Geneva, Switzerland
Stephan Hengstler, MeshEye Consulting, USA
Haowei Liu, Intel Corp, USA
Jin Guohua, Advanced Micro Devices - Boxborough, USA
Yannick Naudet, Public Research Centre Henri Tudor (CRP Henri Tudor) - Luxembourg-Kirchberg, Luxembourg
Theodor D. Popescu, National Institute for Research & Development in Informatics - Bucharest, Romania
Patrick Senac, ISAE (Institut Supérieur de l'Aéronautique et de l'Espace) - Toulouse, France
Euthimios (Thimios) Panagos, Applied Communication Sciences, USA
Christophe Guéret, Vrije Universiteit Amsterdam, The Netherlands

EMERGING Publicity Chairs

Ines Ben Jemaa, INRIA, France
Ken Katsumoto, Osaka University, Japan
Stefan Frey, Hochschule Furtwangen University, Germany

Maarten Wijnants, Hasselt University - Diepenbeek, Belgium
Zhihui Wang, Dalian University of Technology, China
Eric Veith, Wilhelm Büchner Hochschule, Germany

EMERGING 2014

Committee

EMERGING Advisory Chairs

Michael D. Logothetis, University of Patras, Greece
Tulin Atmaca, IT/Telecom&Management SudParis, France
Carl James Debono, University of Malta, Malta
Robert Bestak, Czech Technical University in Prague, Czech Republic
Zoubir Mammeri, IRIT - Toulouse, France
Raj Jain, Washington University in St. Louis, USA
Phuoc Tran-Gia, University of Wuerzburg, Germany
Norihiko Yoshida, Saitama University, Japan
António Nogueira, DETI-University of Aveiro/Instituto de Telecomunicações, Portugal
Ioannis Moscholios, University of Peloponnese, Greece
Henrik Karstoft, Aarhus University, Denmark
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Anne James, Coventry University, UK
Anna Medve, University of Pannonia, Hungary
Nikolaos Tselikas, University of Peloponnese, Greece
Jelena Zdravkovic, Stockholm University, Sweden
Rolf Drechsler, University of Bremen/DFKI, Germany
Christian Blum, IKERBASQUE - Basque Foundation for Science University of the Basque Country, Spain

EMERGING Industry/Research Chairs

Robert Foster, Edgemount Solutions - Plano, USA
David Carrera, Barcelona Supercomputing Center (BSC) / Universitat Politecnica de Catalunya (UPC), Spain
Preetha Thulasiraman, Naval Postgraduate School - Monterey, USA
Anastasiya Yurchyshyna, University of Geneva, Switzerland
Stephan Hengstler, MeshEye Consulting, USA
Haowei Liu, Intel Corp, USA
Jin Guohua, Advanced Micro Devices - Boxborough, USA
Yannick Naudet, Public Research Centre Henri Tudor (CRP Henri Tudor) - Luxembourg-Kirchberg, Luxembourg
Theodor D. Popescu, National Institute for Research & Development in Informatics - Bucharest, Romania
Patrick Senac, ISAE (Institut Supérieur de l'Aéronautique et de l'Espace) - Toulouse, France
Euthimios (Thimios) Panagos, Applied Communication Sciences, USA
Christophe Guéret, Vrije Universiteit Amsterdam, The Netherlands

EMERGING Publicity Chairs

Ines Ben Jemaa, INRIA, France
Ken Katsumoto, Osaka University, Japan
Stefan Frey, Hochschule Furtwangen University, Germany
Maarten Wijnants, Hasselt University - Diepenbeek, Belgium
Zhihui Wang, Dalian University of Technology, China
Eric Veith, Wilhelm Büchner Hochschule, Germany

EMERGING 2014 Technical Program Committee

Adel Al-Jumaily, University of Technology, Australia
Cristina Alcaraz, University of Malaga, Spain
Firkhan Ali Bin Hamid Ali, Universiti Tun Hussein Onn Malaysia, Malaysia
Mercedes Amor-Pinilla, University of Málaga, Spain
Richard Anthony, University of Greenwich, UK
Eleana Asimakopoulou, University of Derby, UK
Tulin Atmaca, IT/Telecom&Management SudParis, France
M. Ali Aydin, Istanbul University, Turkey
Eduard Babulak, Sungkyunkwan University, South Korea
Susmit Bagchi, Gyeongsang National University, South Korea
Zubair Baig, Edith Cowan University, Australia
Kamel Barkaoui, Cedric-Cnam, France
Nik Bessis, University of Derby, UK
Robert Bestak, Czech Technical University in Prague, Czech Republic
Christian Blum, IKERBASQUE - Basque Foundation for Science University of the Basque Country, Spain
Indranil Bose, Indian Institute of Management – Calcutta, India
Kechar Bouabdellah, Oran University, Algeria
Lars Braubach, University *of* Hamburg, Germany
Mieczyslaw Brdys, University of Birmingham, UK
Horia V. Caprita, "Lucian Blaga" University of Sibiu, Romania
Chin-Chen Chang, Feng Chia University - Taichung, Taiwan
Chi-Hua Chen, National Chiao Tung University, Taiwan, R.O.C.
David Chen, University of Bordeaux – Talence, France
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST) - Daejeon, Republic of Korea
Sagarmay Deb, Central Queensland University, Australia
Carl James Debono, University of Malta, Malta
Frank Doelitzscher, Furtwangen University, Germany
Rolf Drechsler, University of Bremen/DFKI, Germany
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Dimitris Drikakis, Cranfield University, UK
El-Sayed M. El-Alfy, King Fahd University of Petroleum and Minerals, Saudi Arabia
Wael M. El-Medany, University of Bahrain, Bahrain
Thaddeus Onyinye Eze, University of Greenwich, U.K.
Kamini Garg, University of Applied Sciences Southern Switzerland - Lugano, Switzerland
Amjad Gawanmeh, Khalifa University, UAE
Nuno Gonçalves Rodrigues, Polytechnic Institute of Bragança, Portugal
Christos Grecos, University of the West of Scotland - Paisley, UK
Christophe Guéret, Vrije Universiteit Amsterdam, The Netherlands
Jin Guohua, Advanced Micro Devices - Boxborough, USA

Sven Hartmann, Clausthal University of Technology, Germany
Go Hasegawa, Osaka University, Japan
Stephan Hengstler, MeshEye Consulting - Campbell, USA
Pao-Ann Hsiung, National Chung Cheng University, Taiwan
Sergio Ilarri, University of Zaragoza, Spain
Muhammad Ali Imran, University of Surrey Guildford, UK
Emilio Insfran, Universitat Politècnica de València, Spain
Shareeful Islam, University of East London, U.K.
Raj Jain, Washington University in St. Louis, USA
Anne James, Coventry University, UK
Ravi Jhawar, Università degli Studi di Milano, Italy
Georgios Kambourakis, University of the Aegean - Samos, Greece
Dimitris Kanellopoulos, University of Patras, Greece
Rajgopal Kannan, Louisiana State University - Baton Rouge USA
Henrik Karstoft, Aarhus University, Denmark
Dimitrios Koukopoulos, University of Patras, Greece
Binod Kumar, Pune University - Jayawant Technical Campus, India
Sy-Yen Kuo, National Taiwan University, Taiwan
Byoungcheon Lee, Joongbu University, South Korea
Mark S. Leeson, University of Warwick - Coventry, UK
Kuan-Ching Li, Providence University, Taiwan
Chiu-Kuo Liang, Chung Hua University, Taiwan
Erwu Liu, Tongji University, China
Haowei Liu, Intel Corp, USA
Li Liu, Utah Valley University, USA
Lu Liu, University of Derby, UK
Michael D. Logothetis, University of Patras, Greece
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Prabhat K. Mahanti, University of New Brunswick - Saint John, Canada
Ahmed Mahdy, Texas A&M University-Corpus Christi, USA
Moufida Maimour, Lorraine University - Nancy, France
Zoubir Mammeri, IRIT - Toulouse, France
Anna Medve, University of Pannonia, Hungary
Natarajan Meghanathan, Jackson State University, USA
Vojtech Merunka, Czech University of Life Sciences in Prague and Czech Technical University in Prague, Czech Republic
Martin Molhanec, Czech Technical University in Prague, Czech Republic
Juan Pedro Muñoz-Gea, Universidad Politécnica de Cartagena, Spain
R. Muralishankar, CMR Institute of Technology - Bangalore, India
Yannick Naudet, Public Research Centre Henri Tudor (CRP Henri Tudor) - Luxembourg-Kirchberg, Luxembourg
Rasha Osman, Imperial College London, UK
Alexander Paar, TWT GmbH Science & Innovation, Germany
Euthimios (Thimios) Panagos, Applied Communication Sciences, USA
Theodor D. Popescu, National Institute for Research & Development in Informatics - Bucharest, Romania
Chuan Qin, University of Shanghai for Science and Technology, China
Fernando Ramos, University of Lisbon, Portugal
Mubashir Husain Rehmani, COMSATS Institute of Information Technology, Pakistan

Copyright Information

Table of Contents

Context Dependent Messages in Delay/Disruption/Disconnection Tolerant Networks

Koki Taguchi, Andrii Zhygmanovskyi, Noriko Matsumoto, Norihiko Yoshida
Graduate School of Science and Engineering
Saitama University
Saitama, Japan
Emails: {taguchi, andrew, noriko, yoshida}@ss.ics.saitama-u.ac.jp

***Abstract*—Delay/Disruption/Disconnection Tolerant Networks (DTNs) attract a great deal of attention as the part of the evolution of various telecommunications networks. In DTNs, the desired features of communication are high-speed data-transfer, high rate of transmission, and reduction of terminal resource usage (e.g., network bandwidth or capacity of node storage). These features often conflict with each other. Although balancing conflicting requirements are important, optimum methodology is still not established. In order to address these issues, we propose a new method which propagates information based on the context of messages in the shared network, such as Vehicle-to-Vehicle (V2V) communication. In our method, we use two parameters, namely, rate of dissemination and maximum number of hops, to control the speed of information propagation and total amount of information in the network. We found that rate of dissemination controls the speed of spreading information and an amount of messages needed for spreading, while maximum number of hops controls the speed of spreading information and the information volume sent per time unit.**

Keywords*—*Delay/Disruption/Disconnection Tolerant Networks; Vehicle-to-Vehicle Communication; Context Awareness

I. Introduction

Delay/Disruption/Disconnection Tolerant Networks (DTNs) is a computer network technology which aims to realize a high trust data transmission in the environment where devices are not able to communicate sequentially. DTNs do not assume a fixed wireless infrastructure, where nodes communicate with each other by direct connection or through multi-hop relays. When the terminal cannot connect to the recipient directly, they perform multi-hop relays by using the resources (message storage) of every node encountered on their communication path. DTNs attract a great deal of attention as a network construction technology for a high-speed communication using only mobile devices, in particular, when network infrastructure is damaged by disasters [1].

As an ideal model of DTNs communication, we discuss the approach to construct a network where any message can be delivered with high-speed and high probability. The simplest way to do it is to broadcast every message. It is difficult for the terminal to deliver its message to the recipient directly in DTNs. The terminal expects that somebody delivers the message by passing its copy through other nodes. The more nodes relaying the message exist, the higher are chances that the message is delivered successfully. However, broadcasting a message involves a lot of relaying nodes, so it is not a good policy since the network becomes flooded with messages. Besides, we need to assume that network resources have some restrictions, since DTNs consist of mobile devices only. For example, the broadcast communication leads to the network congestion and, at worst, to the network failure. Hence, communication on DTNs must be performed in a way that reduces the consumption in resources.

Vehicle-to-Vehicle (V2V) networks [2] consist of many moving vehicles which can communicate with each other, so V2V can be regarded as one form of DTNs. For example, Intelligent Transport Systems (ITS) are one of the solutions for road transportation problems [3][4][5]. The most remarkable feature of ITS is that every communication is performed between the unspecified large number of nodes. Every message has no specific recipient, but it is shared by all nodes which could communicate with others. Assuming that ITS aims to share information, we found that according to context, information could be marked either as urgent or important. These two factors affect information propagation.

Based on this analysis, we propose a parameter-based routing method which controls the process of information propagation by using information context. The structure of this paper is as follows: Section II introduces some existing routing techniques in DTNs. Section III shows our proposal. Section IV shows the results of simulation and related considerations. Finally, Section V presents conclusions and future tasks.

II. Related Works

This section introduces some existing routing techniques in DTNs.

Epidemic Routing [6] is the easiest routing technique in DTNs, where copies of messages are sent to adjacent nodes. With a behavior resembling a contagious disease, this method is a simple technique with very high probability of delivering messages. However, most of messages sent by a node do not reach an intended receiver. In order to avoid an overload of network and message storage, more elaborate message propagation methods have been proposed.

Based on the assumption that using historical information estimates future behavior, several methods using historical transmission state have been proposed. PRoPHET [7] uses the delivery predictability calculated from the encounter rate, the encounter time and the delivery success rate. In MaxProp [8], all nodes in the network calculate transmission costs from the number of the past encounters and send own message along the path determined by these costs. However, in a large-scale mobile network, these methods are hard to apply.

In Encounter-Based Routing [9], the total number of messages is set, and when relaying, sender makes copies of

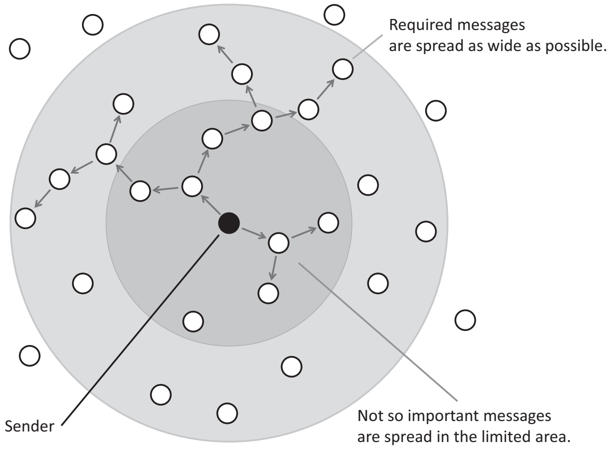

Required messages
are spread as wide as possible.
Sender
Not so important messages
are spread in the limited area.

Fig. 1. Spreading messages by priority.

messages according to the number of node encounters. Spray and Wait [10] introduces two propagation phases. In Spray phase, the fixed number of messages is sent to the network. In Wait phase, the node stops sending messages and waits for them to be received. This policy can limit the amount of information transferred in the network. Spray and Focus [11] differs in that Wait phase is replaced with Focus phase. In Focus phase, nodes holding a message do not wait for it to reach the destination, but pass the message to the other node who can deliver it with higher probability. In these methods, however, the number of nodes who receive a message is limited because the number of messages is fixed.

III. Proposed Method

This study does not target networks for communication between specific communication partners. Instead, it is supposed to be used for sharing information scenarios, such as V2V communication. We propose propagating shared information according to the information context. Assuming that every node wants the information shared, the goal is to spread a message to as many nodes in the network as possible.

We understand that urgency and importance of the information shared in V2V communication are defined by information content. By importance in this paper we mean how much profit can be produced. In case of traffic, a profit could mean that the traffic jam is resolved early, traveling time is saved, and the environment is considered by saving fuel. The priority of such message is high because it is beneficial from a viewpoint of V2V network and the vehicles in the network. Urgency means how much information value is lost as time passes. In other words, this shows how important the freshness of information is. One example of such urgent information is the one that prevents driver's life risk. For example, the submergence of roadway underpass or rockfall along the mountain path. Drivers' safety could be ensured by notifying them of such information in the area as wide as possible.

It is hardly possible to balance all requirements in V2V communication (like yielding both speedy delivery and delivery with high probability), similar to how it is done in general DTNs. Instead, balance should be achieved by considering the information context. Therefore, in this paper, we propose changing propagation method according to a propagation media (in this case, a node), considering the information context to attain desired efficiency (Fig. 1).

A. *Context-Based Communication*

We define context-based communication as follows:

1) Get message context
2) Assign propagation parameters determined by context (importance and urgency) to each message
3) Send information depending on propagation parameters

This paper proposes spreading messages by using parameters (the item 3 above).

During the evaluation (Section IV), we assumed that nodes have functions to get the context and to assign parameters (items 1 and 2 above). Since the number of message types is limited, assigning parameters to each message can be easily performed with the assignment table. It is important to define how the context affects propagation parameters. For example, first, a user sets the level of urgency and importance individually. The level ranges from 0 to 5. Next, the propagation parameters are determined by the level. For example, the level multiplied by 1/5 (20%) is used as rate of dissemination.

B. *Communication Parameters*

The following parameters are proposed:

1) Rate of dissemination: This is a propagation parameter determining a probability, at which a node sends a message to the other node. It controls the number of messages in the network and the speed of propagation. For instance, a sender sends a message to each receiver with 60% probability when a parameter value is 60. In such case, we expect that this information will reach 60% of possible receivers (Fig. 2). This parameter should be set high when the message owner wants to reach more nodes quickly and wants other nodes to relay (the urgency is high).

2) Maximum number of hops: This is a propagation parameter determining the number of intermediate nodes. It controls the area of spreading and the speed of propagation. Basic behavior is similar to Time To Live (TTL) parameter of an

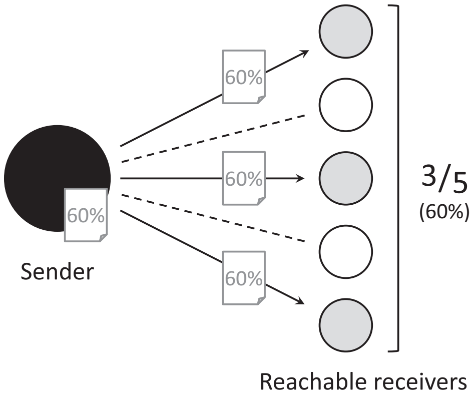


Fig. 2. Example—rate of dissemination (60%).

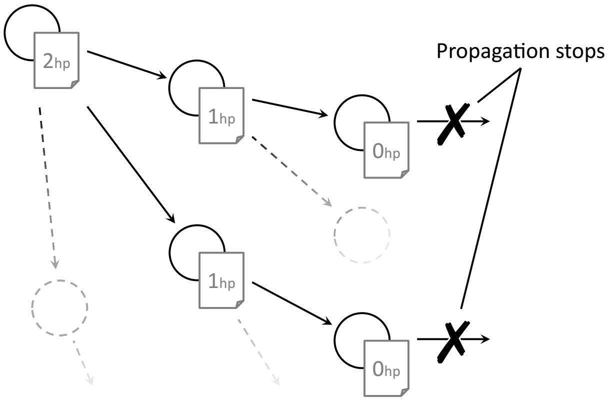


Fig. 3. Example—maximum number of hops (2 hops).

IP packet. Fig. 3 shows an example when the message with maximum number of hops = 2 is propagated. The parameter is decreased by 1 with each hop. The message whose maximum number of hops becomes 0 is not propagated anymore. In case the same information arrives, the message is dismissed. We decided not to overwrite maximum number of hops, since our network aims to spread the information as much as possible. This parameter should be set high when the message owner wants to deliver a message in an area as wide as possible (the importance is high).

IV. Evaluation

Evaluation was done under the assumption that the context of messages (importance and urgency) affects propagation parameters (rate of dissemination and maximum number of hops), which, in turn, influences spreading speed and the amount of information. We use custom DTN simulator written in Java, since none of existing open source software or commercial software met our specific needs for this evaluation. Physical parameters (e.g., communication bandwidth, communication channel or communication delay) were not taken into account.

A. Simulator Design

One cycle of simulation contains the following sequence of actions.

1) Each node moves within the simulation field area.
2) Each node searches other nodes in range.
3) Each node sends messages to nodes found.

All nodes move by 1 unit of length every cycle. The contactable distance of all nodes is 10. The movable field size is 1000 x 1000. To evaluate the effect of our parameters more precisely, nodes and communication relay devices were designed as follows:

- Unlimited storage: prevents deleting messages in the storage.
- Fixed number of nodes: prevents losing messages.
- Communication always succeeds: prevents communication failures affecting successfulness of propagation.

TABLE I. Average Number of Contactable Nodes

Nodes	Upper	Average	Lower
50	0.33	0.02	0.00
100	0.80	0.03	0.00
200	1.18	0.06	0.00

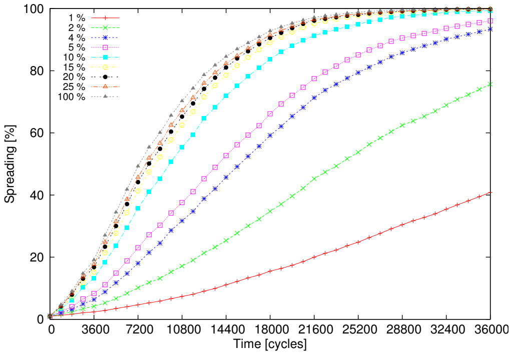


Fig. 4. Spreading (maximum number of hops = 2).

- Free movement: all nodes can either go straight or turn randomly. In an epidemic broadcasting in DTNs, the difference in mobility affects spreading messages [12]. The more freely nodes can move, the more difficult message spreading patterns become.

All shared messages have two propagation parameters and are transmitted according to them. At the beginning of each simulation, the number of messages is the same. The number of messages is affected only by our proposed parameters.

B. Results and Considerations

During the pre-experimentation stage, we measured the average number of contactable nodes using our simulator (Table I). Most nodes did not have communication partner. The more nodes in the simulation field are, the more communication partners for each node appear, since node density increases.

In case of 50 and 200 nodes, an increase in the number of nodes advanced the termination of spreading messages. The simulation produced the same results in case of 100 nodes as to the effect of rate of dissemination and maximum number of hops. Therefore, we considered only the case of 100 nodes. When maximum number of hops is “Unlimited” or rate of dissemination is 100%, given parameter does not have any effect.

Figs. 4 and 5 indicate how long does it take for the message to spread within the field by changing rate of dissemination. The horizontal axis shows the time, and the vertical shows the spreading. By spreading in this evaluation we mean how many nodes receive the message. As nodes in the field receive a message, spreading approaches 100%. Each line corresponds to messages with different values of rate of dissemination. As rate of dissemination increases, the message is spread faster. In other words, rate of dissemination controls the spreading speed. Besides, by comparing two figures, we can see that changing maximum number of hops also affects message spreading speed.

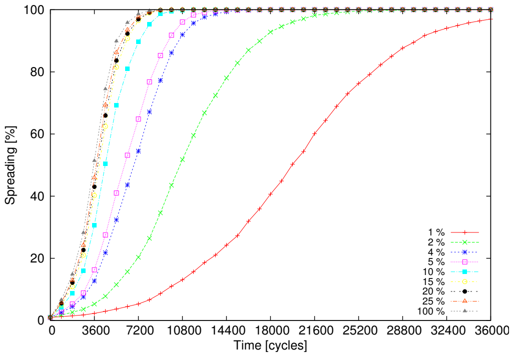


Fig. 5. Spreading (maximum number of hops = 16).

Fig. 6 indicates the number of transmissions for each value of rate of dissemination. The horizontal axis shows the number of cycles. We calculated the average every 600 cycles. Regardless of rate of dissemination, at first the number of transmissions increases with the number of cycles. After that, it stabilizes. The lower rate of dissemination is, the lower is the upper limit of measured value. Fig. 7 indicates the number of transmissions for each value of maximum number of hops. The axes are same as in Fig. 6. As the parameter increases, the number of transmissions increases as well. However, further increase in parameter did not make a substantial change in the number of transmissions. There was a small variation in case of 200 nodes, which we believe is because a small number of nodes restricts the number of hops.

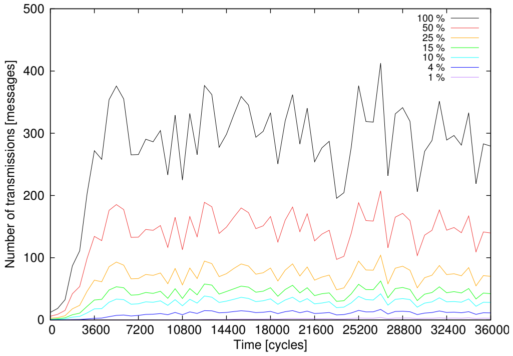


Fig. 6. The number of transmissions (maximum number of hops is fixed).

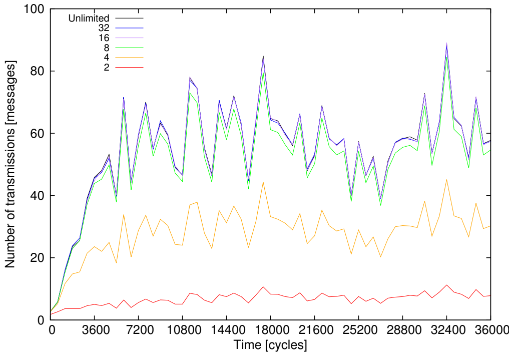


Fig. 7. The number of transmissions (rate of dissemination is fixed).

Fig. 8 shows the number of cycles needed for 100% spreading and the total amount of sent messages. The horizontal axis shows rate of dissemination. The left vertical axis corresponds to the line graph and shows the number of cycles needed for 100% spreading. The right vertical axis corresponds to the bar graph and shows the number of sent messages by the spreading termination. As rate of dissemination increases, cycles until termination amount decreases at first, and then stabilizes. In contrast, the total number of messages increases. Increasing the parameter not only do not expedite the termination, but also generates excessive messages, which, in turn, increase the network load. In Fig. 9, we replaced the horizontal axis to show maximum number of hops. Increasing the parameter decreases cycles until termination amount, similar to rate of dissemination. However, maximum number of hops do not affect cycles until termination amount. Considering Figs. 7 and 9 together, we see that this parameter does not change the sum of messages by spreading termination. From another point of view, it can save network bandwidth by delaying the consumption of network resources at the cost of reducing spreading speed.

According to the experiments above, we assume that proposed parameters (rate of dissemination and maximum number of hops) control the amount of messages in the network and the speed of the message spreading. When rate of dissemination is 50%, the number of messages is reduced by half while the termination time remains almost the same compared to broadcasting. By decreasing the rate to 15%, the number of

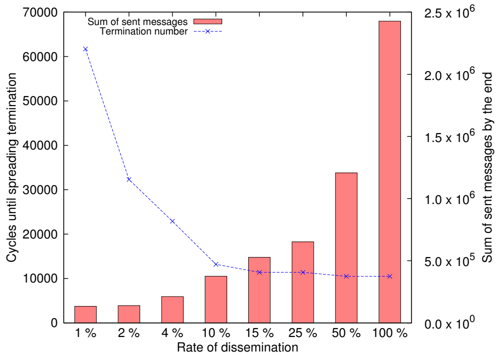


Fig. 8. Cycles until spreading termination and sum of sent messages by the end of the simulation (maximum number of hops is fixed).

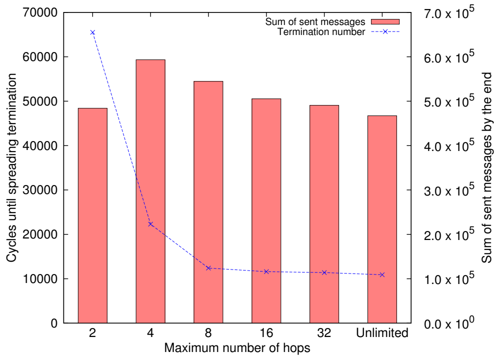


Fig. 9. Cycles until spreading termination and sum of sent messages by the end of the simulation (rate of dissemination is fixed).

messages becomes one fifth and the time becomes about one tenth compared to broadcasting.

V. Conclusion and Future Work

In this paper, we proposed an efficient means of communication in information sharing DTNs by using the context of information. We suggested that two parameters, namely, rate of dissemination and maximum number of hops, can serve as a way of controlling information propagation and help establish the process of context-based communication. We found that rate of dissemination controls the speed of spreading information and an amount of messages needed for spreading, while maximum number of hops controls the speed of spreading information and the information volume sent per time unit.

By performing the simulation, we established the effect propagation parametes (rate of dissemination and maximum number of hops) have on spreading messages, therefore making the usage of message context feasible.

The future tasks for constructing context-aware communication in DTNs are as follows.

- Making the system context-aware: All nodes in DTNs should take the context from their own messages and assign the propagation parameters according to the context (the items 1 and 2 in Section III-A). Also, in order to deal with different kinds of information, the system must be generalized.
- Realistic simulation: We should perform simulation on the various mobile models and consider various channel models and protocols (e.g., CSMA/CA) by using the ns (network simulator) [13][14].
- Using the network state: Each node should take into account the network state when propagating own messages. This allows to build network-aware system.
- Summarization of the similar information: Each node should summarize similar information shared in the network. In the case of V2V, several vehicles may generate the information containing the similar message (e.g., traffic jam occurred at almost the same location). By summarizing such information, we could reduce the total number of messages in the network.
- Applying game theory to DTNs for sharing information: DTNs would become simple and intelligent if each node acts for oneself. The goal of this task is to make the function of each node simple and reduce their load.

References

[1] NEC Corporation, “A Development of a Construction Technology for Large Scale Information Dissemination Networks with Only Mobile Devices,” http://jpn.nec.com/press/201312/20131203_01.html (Japanese), December 2013. [retrived: June, 2014]

[2] Y. Toor, P. Mühlethaler, A. Laouiti, and A. de la Fortelle, “Vehicle Ad Hoc Networks: Applications and Related Technical Issues,” IEEE Communications Surveys & Tutorials, volume 10, issue 3, Third Quarter 2008, pp. 74–88.

[3] Intelligent Transportation Society of America, “ITS America,” http://www.itsa.org/. [retrived: June, 2014]

[4] ERTICO, “Ertico ITS Europe,” http://www.ertico.com/. [retrived: June, 2014]

[5] ITS Japan, “ITS Japan,” http://www.its-jp.org/english/. [retrived: June, 2014]

[6] A. Vahdat, and D. Becker, “Epidemic Routing for Partially-Connected Ad Hoc Networks,” Technical Report CS-2000-06, Duke University, April 2000.

[7] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic Routing in Intermittently Connected Networks,” ACM SIGMOBILE Mobile Computing and Communications Review, volume 7, issue 3, July 2003, pp. 19–20.

[8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks,” INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings, April 2006, pp. 1–11.

[9] S. C. Nelson, M. Bakhit, and R. Kravets, “Encounter-Based Routing in DTNs,” IEEE INFOCOM 2009, April 2009, pp. 846–854.

[10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks,” Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking (WDTN ’05), August 2005, pp. 252–259.

[11] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility,” Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW ’07), March 2007, pp. 79–85.

[12] K. Watabe and H. Ohsaki, “Effect of Locality of Node Mobility on Epidemic Broadcasting in DTNs,” Wireless and Mobile Networking Conference (WMNC 2013), April 2013, pp. 1–4.

[13] “The Network Simulator - ns-2,” http://www.isi.edu/nsnam/ns/. [retrived: June, 2014]

[14] ns-3 project, “ns-3,” http://www.nsnam.org/. [retrived: June, 2014]

Resilient Live-Streaming with Dynamic Reconfiguration of P2P Networks

Kazuki Ono, Andrii Zhygmanovskyi, Noriko Matsumoto, Norihiko Yoshida
Graduate School of Science and Engineering
Saitama University
Saitama, Japan
Emails: {ono, andrew, noriko, yoshida}@ss.ics.saitama-u.ac.jp

***Abstract*—Establishing high robustness and resilience against churn or unexpected behavior of users is an important issue in building reliable Peer-to-Peer (P2P) streaming systems. In P2P-based live streaming systems, interruption and network latency in segment delivery are of particular concern. In order to address these problems, we propose a reliable P2P live streaming system which reconstructs its own topology dynamically, by observing the state of neighbor nodes, selecting a predecessor and spare predecessors, and balancing topologies using network motif. Through various experiments, we show that our approach can reconstruct network topologies in case predecessor's disconnection or defection has occurred.**

Keywords*—*Peer-to-Peer Streaming Networks; Resilience; Dynamic Reconfiguration

I. Introduction

Traditional client-server based live streaming systems can be constructed easily and provide good performance, although it usually imposes high deployment and maintenance costs on the owner. P2P live streaming systems attract much attention because of their superior features: cost reduction, flexibility in case of flash crowds, inherent bandwidth, resource scalability, multiple network paths and self-organization. However, there also exist some problems, for example, increased maximum number of hops, concentrated connection or churn [1].

There already exist many proposals addressing these issues, such as selecting topologies [1], scheduling algorithms [2], incentives [3], re-transmission, coding [1] and balancing topologies according to network motif [4]. However, since these methods presuppose static topology during streaming operations, they cannot address the problem of dynamic reconstruction in resilient P2P live streaming systems. Hence, we propose a reliable P2P live streaming system which reconstructs its own overlay autonomously. This proposal consists of three processes: observing the state of neighbor nodes, selecting next parent node and spare nodes, which are called predecessors, and balancing topologies using network motif.

This paper is organized as follows: we describe research background in Section 2, and present the definition of resilience, related works and our approach in Section 3. In Section 4, we describe our system design. Subsequently, we show the evaluation of our method through simulation in Section 5. Finally, Section 6 contains the conclusion and future work.

II. Background

The demand for user-friendly streaming systems for both content consumers and recipients is increasing. In general, a streaming system is constructed based on client-server architecture. It is important to clarify the difference between client-server based and P2P-based streaming systems. First of all, we describe features of these two, and then discuss features of P2P live-streaming systems.

A. Streaming systems

In streaming systems, large files are transmitted, causing the original content to be divided into many small segments. The recipients play back received segments while downloading next segments. Streaming systems can be divided into two types: client-server based systems and P2P-based systems. Furthermore, these systems are classified by distribution method: Video-On-Demand (VOD) and Live. In VOD steaming systems, all clients that participate in the streaming network can play segments at any time. However, in live streaming systems, all clients need to play the same segments at the same time. On that account, constructing live streaming systems based on P2P is more difficult than doing it based on client-server approach. Therefore, we focus on P2P-based live streaming systems.

1) Client-server based streaming systems: In client-server based streaming systems [5], all clients request desired content from the server who owns them. If the number of clients increases significantly, the server becomes heavy loaded. Moreover, in the worst case of server down, all clients will be unable to download any content at all. There are some methods that address these issues, the simplest one being to prepare extra servers in advance. This solution incurs high deployment and maintenance costs, although it is a very easy way which performs well. For example, current estimated cost of YouTube [5] is 1 million dollars per day [1].

2) P2P-based streaming systems: It is necessary to note that unlike client-server based streaming systems, all clients in P2P-based streaming systems [6][7] can be both senders and receivers. In general, P2P-based systems are not as costly as client-server based systems, because the former can distribute the load of server using all resources of participating nodes. Therefore, various P2P streaming systems were proposed to perform Live and Video-On-Demand streaming to mass audience with better quality at low server and deployment costs. While having these advantages, P2P live streaming has also the drawback of network topology becoming highly dynamic because of churn. Therefore, interruption of streaming and latency in segment delivery are frequently observed in P2P live streaming systems.

In order to address these problems, it is important to ensure that network is resilient. A definition of resilience is given by

Abbound et al. [1] as follows: “The persistence of avoiding too frequent or severe failures in the presence of changes.” We show some approaches that ensure resilience in Section 3.

B. P2P Live-Streaming

As mentioned above, there are some problems in P2P live streaming systems. They can be roughly divided into the following three kinds:

1. **Highly dynamic topology**
   P2P-based streaming systems let all clients decide freely when to join or leave. This results in frequent topology changes.
2. **Strict real-time constraints**
   In live streaming, all nodes must ensure synchronous playback. However, in P2P live streaming systems, the number of hops differs for each node. Therefore, it becomes difficult to accomplish synchronous playback.
3. **Topology imbalance**
   In P2P networks, all nodes select a predecessor randomly. This causes an increase in the number of hops and connection concentration.

In order to address these problems, especially the dynamic change of network topologies as well as strict real-time constraints, the system needs to ensure resilience.

III. Resilience in P2P streaming systems

To ensure resilience in P2P live streaming systems, different approaches has been proposed. In this section, first we explain several of them, then introduce the related work that concerns balancing topologies using network motif. Finally, we introduce our approach.

A. Approaches to ensure resilience

There are the following approaches to ensure resilience in P2P live streaming systems:

1. **Topology selection**
   There are three basic topologies: a tree, a mesh, and a hybrid of them. Topology selection should be based on actual load, streaming interruptions and amount of effort needed to construct an overlay among others.
2. **Scheduling algorithms**
   In P2P streaming systems, each node has different performance. To maximize the quality of content and reduce end-to-end delay, a system should determine which segments should be re-transmitted and when these transmissions should be carried out.
3. **Re-transmission**
   Since a node can request a missing packet from another node (not necessary being its original sender), re-transmission provides resilience. However, in live streaming, the node must receive missing packet before proceeding with current stream playback. Therefore, this approach has strict time constraints.
4. **Incentives**
   System performance may suffer significantly if nodes do not contribute their resources fully, for example, restricting upload bandwidth or leaving the system once segments have been delivered. Therefore, the system should encourage nodes to stay connected.
5. **Network coding**
   Network coding is a technique by which nodes in a streaming system encode multiple blocks into one instead of simply sending data blocks. This approach leads to decreasing the amount of packets in the network. However, this algorithm can be difficult to implement.
6. **Media coding**
   Media coding enhances resilience by allowing the receiver to deal better with losses and bandwidth fluctuations.

B. Network motif

In P2P live streaming systems, topology imbalance poses a serious problem. In order to address this problem, a method of balancing topologies using network motif is proposed [4].

1) Network Motif: Network motif is defined as recurrent of statistically significant subgraph or pattern. The aim of this concept is to narrow the gap between local and global knowledge of large networks and to understand network structure better. In network motif, each node has at least one input or output.

Network motif can be used as an approach to compare the difference of network characteristics in biological studies, World Wide Web (WWW) or electric circuit networks [8].

2) Approach in related work: In the related work [4], all nodes make request to their predecessors/successors and all transfers are performed from successor to predecessors when they enter the network or are already connected to successors. Each node’s behavior is as follows:

1. **When a node enters the streaming network**
   1) Select a predecessor randomly.
   2) A node, which enters the streaming network, obtain immediate predecessor’s information and predecessor’s parent information.
   3) Reconnect to a new predecessor that has sufficient load margin.

2. **When a node is connected to successors**
   1) Get neighbor node’s information.
   2) Compare neighbor node’s load with one’s own.
   3) Reconnect a successor to a new predecessor that has sufficient load margin.

This approach makes use of balanced tree topologies, decreases the server load and increases the scalability in case of large audience. However, this exchange operation is effective only when nodes enter a streaming network or are already connected to successors, and it does not consider the predecessor’s disconnection or defection. Therefore, it can be said that this approach ensures resilience only in static manner.

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-357-5

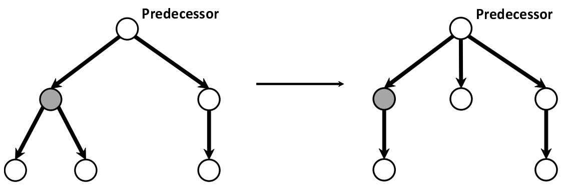


Fig. 1. Situation when predecessor has sufficient margin in load.

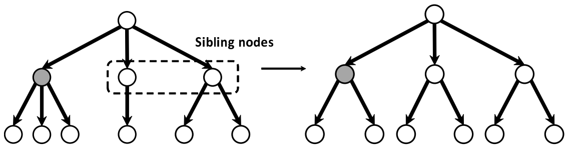


Fig. 2. Situation when sibling nodes have larger margin.

C. Our approach

At present, most live streaming systems are client-server based; but, since this approach incurs high deployment and maintenance costs, P2P live streaming systems draw much attention. However, as mentioned above, there are still many problems that arise in P2P live streaming systems.

In order to address these problems, several approaches [2][3][4] were proposed; however, none of them concern reconfiguring topologies dynamically which is important in P2P networks, and overall there are few works that address such situation. Therefore, in this paper, we propose a new approach and evaluate it by simulation.

IV. System design

In this section, we describe our approach for ensuring resilience by reconfiguring topologies dynamically. This approach consists of three methods: observing neighbor node's state, selecting a predecessor and spare predecessors, and balancing topologies using network motif.

A. Observing neighbor nodes state

Each node observes its own neighbor nodes while playing back current stream. If node's predecessor has sufficient margin in the load, then the node reconnects its successor to that predecessor. If the predecessor has not enough margin in the load but some of its siblings have margin larger than its own margin, then the node reconnects its successor to one of its predecessor's siblings.

Figures 1 and 2 show examples of such a behavior. In these figures, each node is allowed to be connected with at most three nodes.

B. Selecting next predecessor and spare predecessor

Each node is able to continue streaming by quickly replacing its own predecessor in case of disconnection or defection of the latter. Exchange operations during replacing of predecessor are as follows:

1) Each node observes its predecessor's state: normal, disconnected or defected.

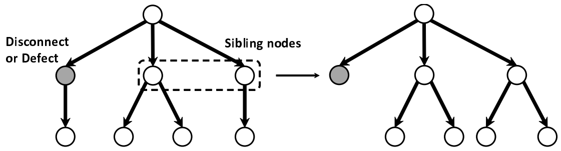


Fig. 3. Replacing predecessor.

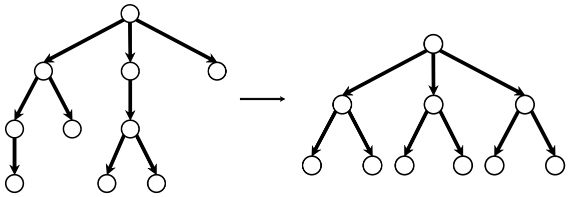


Fig. 4. Balancing topologies using network motif.

Algorithm 1 Pseudo code of our approach

```
Input: $Neighbor_i, Parent, Child$;
  while playback segments do
    $ObserveNeighbor(Neighbor_i)$;
    $NewParent \Leftarrow Neighbor_1$;
    $SubParent \Leftarrow Neighbor_2$;
    if $NumChilds > Neighbor'_isNumChilds$ then
      $Reconnect(Child, Neighbor_i)$;
    end if
    if Parent is Disconnect or Defect then
      if NewParent is Alive then
        $Parent \Leftarrow NewParent$;
      else
        $Parent \Leftarrow SubParent$;
      end if
    end if
  end while
```

2) A node selects new predecessor from predecessor's sibling nodes by largest margin in the load.
3) In case of predecessor's disconnection or defection, each node reconnects to the new predecessor.

Figure 3 shows an example of this behavior.

C. Balancing topologies using network motif

P2P live streaming is prone to topology imbalance resulting in an increase in the number of hops as well as connection concentration. Therefore, in this work we adopt a topology balancing approach that was proposed by Krumov et al. [4].

Using this approach, we achieve both an optimization of hop count and static load balancing in each node.

Figure 4 shows an example of balancing topologies, and Algorithm 1 presents a pseudo code of our approach.

V. Evaluation

In this section, we present evaluation results of our approach. They include structural properties of produced topologies and dynamic reconfiguration.

TABLE I. SIMULATION PARAMETERS.

Parameter	Value
Number of predecessors	1
Number of successors	3
Number of original senders	1
Number of pseudo segments	5
Playback time for pseudo segment	5000 ms
Simulation time	25000 ms

We compare “Normal Approach”, in which each node in a streaming topology connects to a predecessor randomly, with “Proposed Approach”, which corresponds to our approach. In all of our experiments, we use the following scenario: there is one root node providing the system with the original streaming signal.

We investigate our approach from three different perspectives: topology’s height and latency, connectivity and exchange steps for each node, and Topo-metric values.

For each node, Topo-metric value is defined in [4] as the difference between the longest and the shortest branches of succeeding subtrees, starting from certain node in the whole tree. If Topo-metric value is large, the streaming topology becomes imbalanced. Therefore, smaller value is preferrable.

To evaluate our approach, we prepared a simulator which is executed on single computer. Some major parameters of the simulation are summarized in Table I.

A. Height and latency in produced topologies

Figure 5 shows the height of produced topologies, and Figure 6 shows their latency in a range from 10 to 100 nodes. In this experiment, we did the following actions randomly: addition of a new node, predecessor’s disconnection or predecessor’s defection. These results show the average of 10 executions for each experiment.

Figure 7 shows the change in topology’s height in case predecessor’s disconnection or defection has occurred. In this figure, these events occurred at approximately 2000, 10000 and 16000 milliseconds after the simulation had started.

From these results, we can conclude that both the height and latency in produced topologies are reduced using our approach. Furthermore, the height of topology increases after predecessor’s disconnection or defection, although it gradually decreases afterwards. Therefore, our approach can reconfigure the topologies dynamically in case of predecessor’s disconnection or defection, and decreases maximum number of hops.

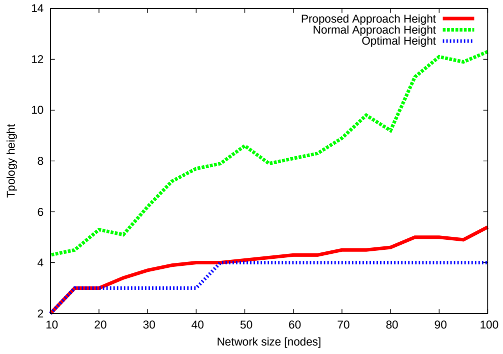


Fig. 5. Height of streaming topologies.

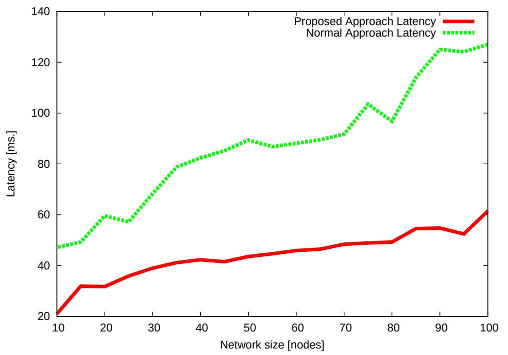


Fig. 6. Latency of streaming topologies.

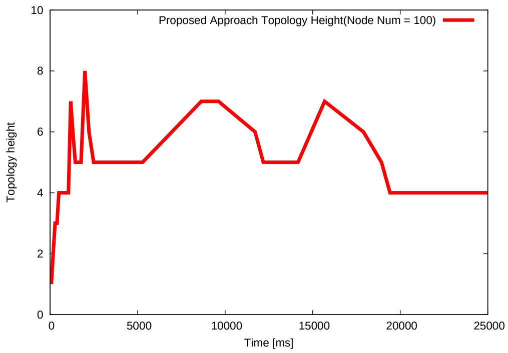


Fig. 7. Change of topology’s height in time.

B. Number of exchange steps and connectivity in produced topologies

Connectivity is the number of nodes to which a node is connected. In this work, each node allows up to 3 connected nodes, therefore theoretically optimal node connectivity is exactly 3.

Figure 8 shows the average connectivity for each node in produced topologies, and Figure 9 shows the number of exchange steps for each node in produced topologies, in a range from 10 to 100 nodes. In this experiment, we did the following actions randomly: addition of a new node, predecessor’s disconnection or predecessor’s defection. These results show the average of 10 executions for each experiment.

Figure 10 shows the result of a dynamic change of the average connectivity for each node in case predecessor’s disconnection or defection has occurred. In this figure, these events occurred at approximately 2000, 10000 and 16000 milliseconds after the simulation had started.

From these results we conclude that with our approach amount of successors for each node increases regardless of the network size. The average of exchange steps per node is also independent of the network size, and in total it grows

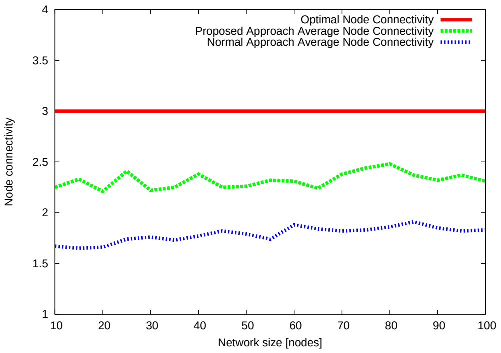


Fig. 8. Average node connectivity.

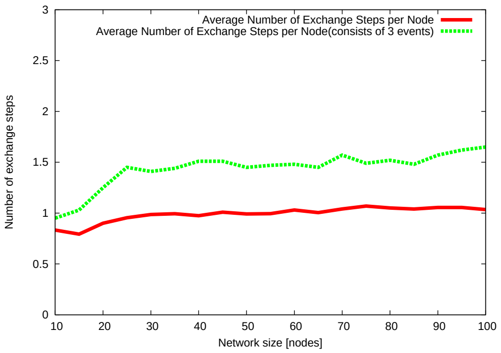


Fig. 9. Number of exchange steps per node.

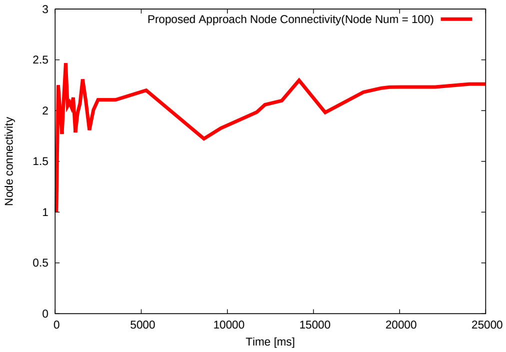


Fig. 10. Change of average node connectivity in time.

sublinearly with respect to network size, increasing from 1 to more than 1.5.

Furthermore, the connectivity of each node decreases after predecessor's disconnection or defection has occurred, however, it gradually increases afterwards. Therefore, our approach can reconfigure the topologies dynamically in case of predecessor's disconnection or defection, and achieves dynamic load balancing.

C. Topo-metric values in produced topologies

Figure 11 shows the Topo-metric value in produced topologies, in a range from 10 to 100 nodes. In this experiment, we did the following actions randomly: addition of a new node, predecessor's disconnection or predecessor's defection. These results show the average of 10 executions for each experiment.

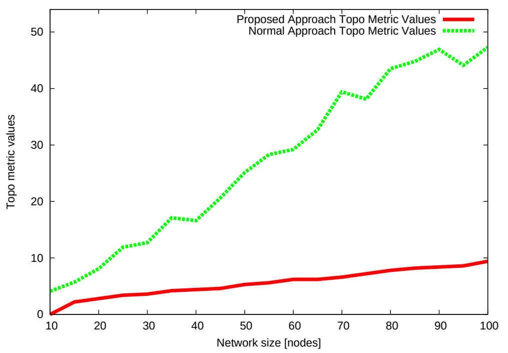


Fig. 11. Change of Topo-metric value wuth regard to network size.

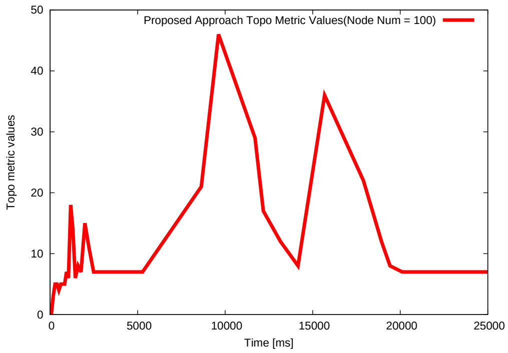


Fig. 12. Change of Topo-metric value in each time.

Figure 12 shows the result of a dynamic change of the Topo-metric value in case predecessor's disconnection or defection has occurred. Here, these events occurred at approximately 2000, 10000 and 16000 milliseconds after the simulation had started.

From these results, we can conclude that Topo-metric value in produced topologies is lowered using our approach. Furthermore, the Topo-metric value increases after predecessor's disconnection or defection, although gradually decreases afterwards. Therefore, our approach can reconfigure topologies dynamically in case of predecessor's disconnection or defection, and produces balanced topologies.

D. Change of Topo-metric values in produced topologies

In Figures 13 and 14, we have fixed two positions in the network where disconnection or defection occurs, and labeled them as low and high. In case of a low position, node is located 1 to 2 hops up from the leaf node of a tree, and in case of a high position node is located 1 hop down from the root node. In these figures, both events occurred at 6000 milliseconds after the simulation had started, and streaming topology consists of 100 nodes.

From these results, we can see that if a lower node was selected, the time to stabilize the Topo-metric value becomes shorter compared to selection of a higher node. This result is the same regardless of whether disconnection or defection has

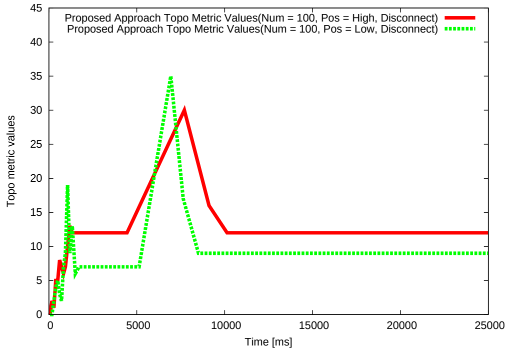


Fig. 13. Change of Topo-metric values in time (in case of predecessor's disconnection).

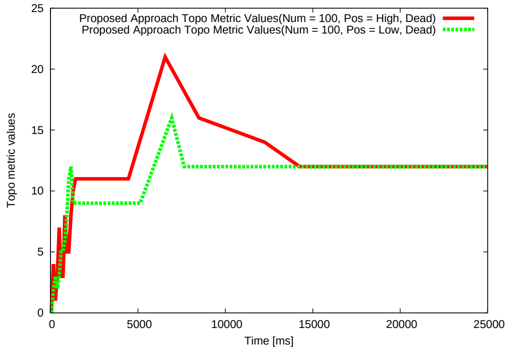


Fig. 14. Change of Topo-metric values in time (in case of predecessor's defection).

occured. From this we can conclude that the imbalance in the topology tends to appear if a higher node is disconnected or defected.

VI. Conclusion and Future work

This paper studied the resilience of P2P live streaming systems that comes as a result of dynamic reconfiguration of peer-to-peer network.

In this study, we considered the scenario which consists of a source node that provides the original streaming signal, and other nodes in topology, that distribute the signal among each other. In this scenario, our approach greatly decreases height and latency in produced topologies, and provides an ability to dynamically reconfigure the network in case of predecessor's disconnection or defection.

Issues to address in further studies are as follows.

A. *Dealing with free-riders and malicious nodes*

Our simulator does not consider free-riders or malicious nodes: all nodes in the streaming network receive segments and send them to successors. However, in real P2P live streaming systems not all nodes are like this. Even if free-riders and malicious nodes are present, it is difficult to detect them, and the efficiency of load balancing using our approach suffers. In order to address this problem, Simple Trust Exchange Protocol (STEP) is advocated as a possible solution [9]. Therefore, we need to improve our approach and simulator to model actual P2P live streaming systems.

B. *Practical experiments using HTTP Live Streaming (HLS)*

Our simulator was executed on single computer, and lacks the following features: recording, encoding and segmentation. Therefore, the load applied on root node in our simulator is much smaller compared to real world scenario. Furthermore, since we used pseudo segments in our simulator, the latency is not exact. Because of these issues, we need to measure latency and load rigorously when segments are delivered to successors. In order to address these problems, we need to consider using HTTP Live Streaming [10], and confirm the soundness of our approach.

References

[1] O. Abbound, K. Pussep, A. Kovacevic, K. Mohr, S. Kaune, and R. Steinmetz, "Enabling Resilient P2P Video Streaming: Survey and Analysis", Multimedia System, June, 2011, pp. 177–197.

[2] Y. Guo, C. Liang, and Y. Liu, "AQCS: Adaptive Queue-Based Chunk Scheduling for P2P Live Streaming", Proceedings of 7th International IFIP-TC6 Networking Conference Singapore, May, 2008, pp. 433–444.

[3] Z. Liu, Y. Shen, S. S. Panwar, K. W. Ross, and Y. Wang, "Using Layered Video to Provide Incentives in P2P Live Streaming", Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV, August, 2007, pp. 311–316.

[4] L. Krumov, A. Andreeva, and T. Strufe, "Resilient Peer-to-Peer Live-Streaming using Motifs", IEEE WoWMoM, June, 2010, pp. 1–8.

[5] YouTube, http://www.youtube.com/. [retrived: June, 2014]

[6] BitTorrent, http://www.bittorrent.com/. [retrived: June, 2014]

[7] J. Xiong and R. R. Choudhury, "PeerCast: Improving Link Layer Multicast through Cooperative Relaying", IEEE INFOCOM, April, 2011, pp. 2939–2947.

[8] C. Brandt and J. Leskovec, "Status and friendship: mechanisms of social network evolution", WWW Companion '14 Proceedings of the companion publication of the 23rd international conference on World wide web companion, April, 2014, pp. 229–230.

[9] Y. Yoshida, M. E. Haque, N. Matsumoto, and N. Yoshida, "Efficient Decentralized Evaluation of Node Trustworthiness in Peer-to-Peer Networks", Proceedings of International Conference on Computer Engineering and Technology 2009, July, 2009, pp. 177–179.

[10] HTTP Live Streaming Resources - Apple Developer. https://developer.apple.com/streaming/. [retrived: June, 2014]

ISBN: 978-1-61208-357-5

Cloud Resource Price System

Sururah A.Bello
Computer Science & Engineering Department
Obafemi Awolowo University
Ile-Ife, Nigeria
sbello@oauife.edu.ng

Claudia Lüthje, Christoph Reich
Cloud Research Lab
Furtwangen University
Furtwangen, Germany
{claudia.luethje, christoph.reich}@hs-furtwangen.de

***Abstract*—Cloud Resource pricing has been known to be static. The rapid nature of resource deployment gave the impression that differential pricing might not be possible. This study shows how various factors (e.g., Social User Status, Cloud Provider Reputation, SLA Type, etc.) can be greatly used to achieve a dynamic pricing system in Cloud provisioning. An architecture of a resource price expert system has been developed, that combines mathematical relations, IF-clauses and Neural Networks to achieve a resource price model for cloud systems. Based on use cases the effectiveness of this solution is shown.**

Keywords—Cloud Computing; Price Models; Expert System

I. Introduction

No doubt, the PC and the Internet brought about an information revolution by making information universally accessible and affordable.

Cloud computing is a computation revolution, that gives users the possibility to access and utilize massive amounts of processing power and computer resources [1]. For example Amazon offers currently a machine with 613MB RAM [2] for 2 cents per hour, and there exist even cheaper offers from other providers. These prices are very small compared to the cost of a computer system. Thus cloud computing enables large computation affordable and universally accessible.

However, the computation revolution has to be realized with appropriate business models that will make the economic case prevail and thus make cloud computing a consumer commodity. Payment for Internet services has traditionally favored the flat pricing mechanism [3], which is basically monthly and lately hourly subscription (pay-per-use). According to [4], this static model of pay-per-use and subscription pricing allows easy prediction of payments. However Lai [5] found that dynamic pricing policies could achieve more economically efficient allocations and prices for high-value services. The ability of cloud providers to attract more cloud users by employing dynamic pricing through the offering of customized pricing models for the same product for different customers could translate into more income for the providers.

In this paper, a new concept of building a pricing model will be shown. Also the use of additional fuzzy information about the customer will be investigated as input parameters for the fair and economical price model. To create such an dynamical pricing model, artificial intelligence offers various techniques. An evaluation of neural networks has brought out to be the best solution. This method has been proven to be successful in the past.

After discussing related work in the next section, the remainder of this paper is structured as following. Section III will discuss what is influencing the price of a cloud resource in general. Section IV gives a detailed description of the architectural design of the price module, which is evaluated by use cases in section V. In section VI, a conclusion is drawn.

II. Related Work

The current static pricing in Cloud Computing basically employs consumption patterns for pricing i.e., a fixed price for a fixed quantity per hour. Even though a number of variations are being introduced, like Amazon offering the opportunity to reserve machines in advance with an upfront payment which then grants an discount during usage, the basic model remains static.

Strømmen-Bakhtiar has discovered that there is a tendency for customers to think that paying the same price as in th beginning after a period is not cost effective. Perceived fairness in pricing significantly relates to emotions, and emotions similarly affect behavioral responses. This means perceived unfairness can lead to distrust and diminished shopping intentions both off and on the Internet. When consumers perceive price unfairness they feel negative emotions like anger, outrage, disappointment and may not repeat purchase.

Lai delivered, with an empirical study, evidence that various types of differential pricing tactics can have a significant impact on consumers' perception of price fairness. In addition it has been shown that employing dynamic pricing through offering customized pricing for the same product for different customers could translate into more income [5]. Differential pricing strategy involves charging varying prices for the same product based on some characteristics of the customer or the product.

Miyazaki [7] identified some differential pricing tactics that are available to Internet stores: buyer identification, time of purchase, purchase quantity, and asset/usage. In an other paper [8] this was extended to pricing of Cloud Computing since Cloud Computing is also a business transacted over Internet infrastructure like the Internet stores. No doubt there is a need to develop appropriate business models that will continuously make Cloud Computing more attractive to users. Therefore business models have to be created which transform cloud computing to a pure consumer commodity. In this paper we present a further pricing scheme for a customer specific business model. Against four factors proposed in [8], a number of factors have been considered here to be influential in developing customer centric Business model for Cloud Computing.

III. Cloud Resource Price Influence Factors

Several factors have been identified which influence the price of cloud services. For this a classification in fixed and variable factors was created. As an fixed factor things like the cost of operating the data center can be seen. Examples for variable factors are; Social Category of a User, Cloud Providers Reputation, Type of Service Level Agreement (SLA), the Reputation of a User, Availability of Monitoring Services, Public Review and Type of Co-Cloud Users. An overview of these factors can be seen in Fig. 1. Subsequently, the respective factors are described in more detail.

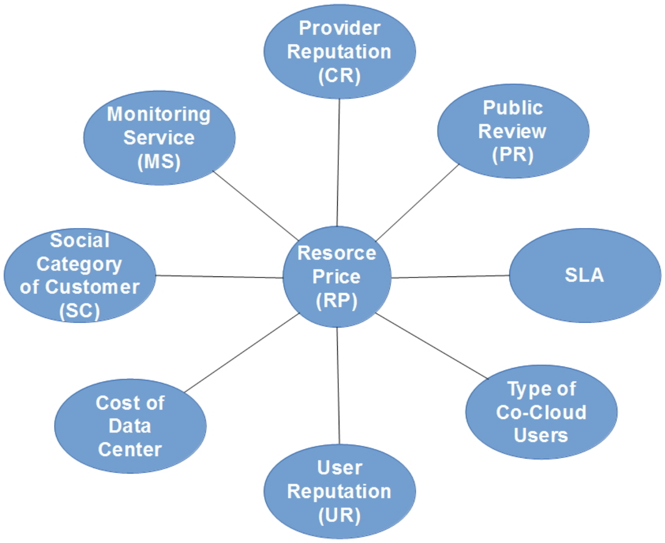


Fig. 1: Factors Influencing the Price of Cloud Resources

A. Cost of Data Center

According to [9], the following entities make up the cost of the data center. The cost of real estate, power from the grid, backup power (generators and batteries) maintenance of backup power, cooling resources, maintenance of cooling resources, security, network connectivity and fire safety. The cost may vary from location to location but are fixed for a particular location.

B. Social Category of Customers (SC)

A fair price should be charged to everyone. But the price should be adjusted for the different needs thus medical doctors may charge different fees to different patients [10]. Thus it is established that a price differential may be on account of the social status of a consumer. Social classifications of users is done here and presented with corresponding pricing scenarios. An adjusted price may be proffered for a cloud user perceived to be in need, in this case classified as a poor user. Classification can be based on the location of the consumer. The pricing of a hotel room in a downtown area might be different from a similar room in a resort. In countries like Nigeria if a company is situated in the nation's capital, it is perceived to be a rich company. Hence location could be used to classify cloud users. The year of operation can also be used to determine the stability hence the ability of a company. Magazine publishers offer price promotions to new subscribers to enhance their purchase intention. So a company that has existed for some years can be seen to be stable hence classified as being rich. Thus the year of operation is employed for social status in this work. A company of less than 2 years is categorized as *New*, those between 2 and 4 years as *Middle* and above 4 years are taken as *Old*.

C. Cloud Provider's Reputation (CPR)

Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another. A single violation of trust can destroy years of slowly accumulated credibility. There are a number of ways to establish online trust [11]. Reputation is a component of online trust [12] and it also measures reliability. Reputation is the belief by the community of an entity's stand. Using Cloud infrastructure for critical business computation necessitate that the reputation of the Cloud provider is well established. A single publicized unethical activity could create uncertainty that could tarnish a longterm reputation. The Amazon botnet is a major compromise of a cloud provider. Company's reputation could be in terms of success rate of transactions [13] and confidentiality of user's data. Record of Cloud providers experience could serve as certificate of credibility for future patronage. Hence, a Cloud Provider's reputation can be greatly used to negotiate prices for cloud services. Provider's Reputation could be quantified using: Record of Past Experience and Record of Compensation in case of problem. This study uses these two to express the reputation of Cloud Providers.

D. SLA

As guarantees for service delivery SLAs are negotiated between the Cloud Providers and Cloud Consumers. Most often SLAs are dictated by the Cloud Providers [14]. Providers usually have fixed templates of SLAs where cloud users are expected to pick from. There is opportunity to also negotiate SLA online as reported in [15], where cloud user could be allowed to specify customized SLA at a price. Therefore the type SLA could be used to influence the price to charge.

E. Users Reputation

The nature of multi-tenancy in Cloud Computing demands that users should have a positive reputation. Sniffing programs, Trojans, Ill motive, Attacker and Hackers from the users end can endanger the Cloud [16]. Social engineering attacks remain effective - one exploit tried to convince Amazon Elastic Compute Cloud (EC2) users to run malicious virtual machine images simply by giving the image an official sounding name, such as “fedora core”. It's apparent that not only the data and software is worth protecting in the Cloud but also the activity patterns. Activity patterns could constitute confidential business information [1]. Users reputation could be quantified with the physical address (UPL), the police records of criminality (UCR) and also users bank details (UBR) of financial stability as banks don't give credit to bankrupt customers. This study considered the three of users reputation.

F. Availability of Monitoring Services (MS)

Few Cloud Providers have the confidence to provide customers with monitoring tools for service availability. Rackspace Inc., GoGrid or ZOHO are offering 24/7 customer support for free. However, Microsoft or Amazon is more inclined to provide paid customer support. Google's attitude is more likely: "Cloud it on your own" towards customers. Independent monitoring is largely missing except for Gomez Inc. and Hyperic Inc. that offer services for monitoring providers, SLA compliance and elasticity. Hence a good means for negotiating customer friendly pricing could be availability of proper monitoring devices. Monitoring services could either be from the provider itself or a third party.

G. User's Recommendation, Feedback and Public Review(PR)

Commoditization of Cloud services must emphasize users rights to have a voice. Public reviews on issues such as downtime, phishing [17], data loss, password weakness can be valuable in pricing of cloud services. User ratings is employed in this study, as done by Airline Operators (from 1 to 5): *1-Excellent*, *2-Very Good*, *3-Good 4-Fair* and *5-Bad*

H. Co-Cloud Users

The nature of multi-tenancy in a Cloud could enable competitive companies to use the same Cloud platform. There may emerge clash of interest, fear of possible leakage of confidential business information, loss of privacy, risk of data theft [12]. Multi-tenants on cloud infrastructure has introduced non-obvious threats as a result of sharing physical resources between VMs [18]. Hence information about co-tenants in the Cloud can be used to influence service price. No matter how cheap the cloud service is, the presence of a business competitor may scare off similar companies. In the same vein, a cloud provider may offer a high cost in exchange for a deal not to host a competitive company.

IV. ARCHTECTURE OF THE CLOUD PRICE MODULE

To get a most variable price solution for the customer and the provider, a number of data has to be collected and evaluated. The information is combined as shown in the following Figure.

There are two major interfaces: The *customer interface* and the *provider interface* to supply the information, from which the price is determined. On both interfaces the parameters are divided into fixed and variable cost parameters. The interface of the customer requires information about which resources the customer requires, which are fix parameters like the storage size, CPU and RAM and variable parameters like the storage place and runtime. Also the provider interface has input values. There the fix costs are CPU and RAM and the variable costs are defined as categories and reputation. This means the variable costs directly depend on the status of a customer. At the *customer interface* different parameters are influencing the price for Cloud. SLA specific parameters are modelled as XML using the Adaptable Service Level Objective Agreement (A-SLO-A) language [15].

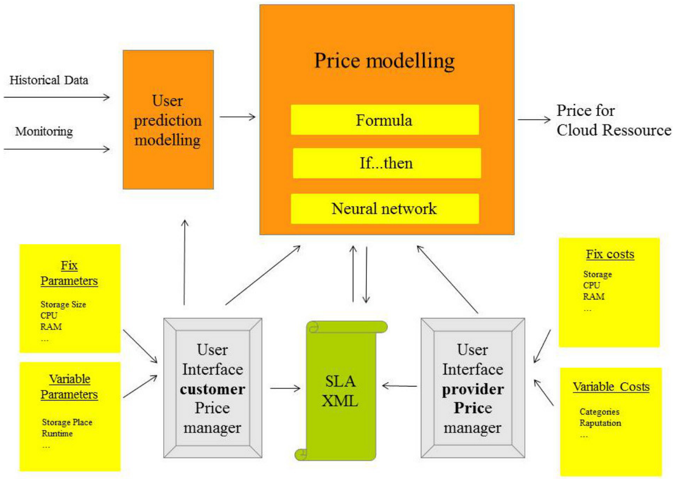


Fig. 2: Architecture of the Cloud Resource Price System

A. Mathematical Model

First of all the fixed costs or parameters are calculated by mathematical formulas. Then the costs based on the input parameters will be added:

- Costs of Power
- Costs of Hardware (CPU, RAM, Storage)
- Costs of required SLA

For example, cost-power + cost-hardware + cost-sla = cost-total

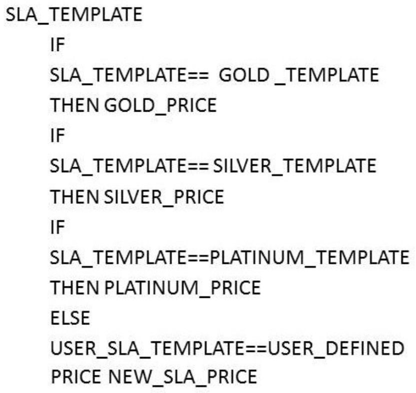
This cumulative costs will be shown to the customer, to support them, to find a suitable solution

B. Rule Based Model

The model is described using IF-THEN rulesets to specify the options available. For the SLA factor, the rule based design is shown in Fig. 3. The SLA level is used to influence price for a service. The provider has three distinct SLA templates for the user to choose from; GOLD_TEMPLATE, SILVER_TEMPLATE and the PLATINUM_TEMPLATE, each of which is accompanied by its price. The user has also the opportunity to negotiate a new SLA, specifically designed for its needs as it has been established that SLA can be negotiated online.

C. ANN Model

The artificial Neural Network Resource Price (RP) system was developed to further illustrate the concept. Four steps were adopted in training of the neural network. First assemble the training data, second create the network neurons, third train the network and fourth run the neural network. The feed-forward neural network with back propagation was used because of their simplicity. The ANN structure is as shown in Fig. 4 with one layer of hidden neurons followed by an output layer. The inputs are Public Review (PR), Monitoring Service (MS), Cloud Provider's Reputation (CPR), Type of SLA (SLA), User Physical Location (UPL), User Bank Record (UBR), User



```
SLA_TEMPLATE
    IF
    SLA_TEMPLATE== GOLD _TEMPLATE
    THEN GOLD_PRICE
    IF
    SLA_TEMPLATE== SILVER_TEMPLATE
    THEN SILVER_PRICE
    IF
    SLA_TEMPLATE==PLATINUM_TEMPLATE
    THEN PLATINUM_PRICE
    ELSE
    USER_SLA_TEMPLATE==USER_DEFINED
    PRICE NEW_SLA_PRICE
```

Fig. 3: Knowledge Represented as Rules

Criminal Record (UCR) and the Social Category of user (SC). As shown in Fig. 5, different combinations of the inputs where used for the training of the neural network.

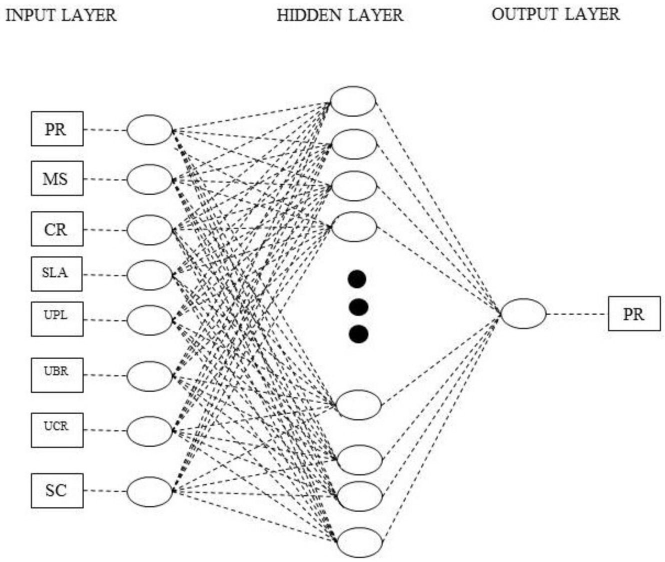


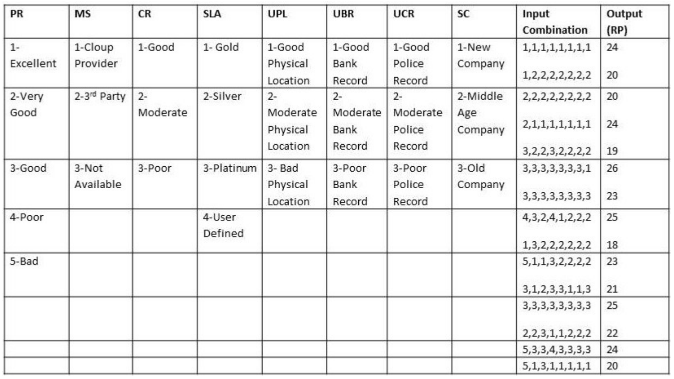
Fig. 4: Knowledge Represented with ANN

The function newff was used to create the ANN. For the initial network, the following command was used and the ANN network object called “CloudNet” was created.

```
CloudNet = newff(minmax(input_combination),
    [100,1],
    {’tansig’,’purelin’},’traingdx’);
```

Also the weights and biases of the network had to be initialized. The following function transfers tansig from input layer to hidden layer and purelin from the hidden layer to output. The training function is used for the back-propagation traingdx. Several trainings were carried out and the CloudNet parameter that gave the best performance is shown below (see Figure 5):

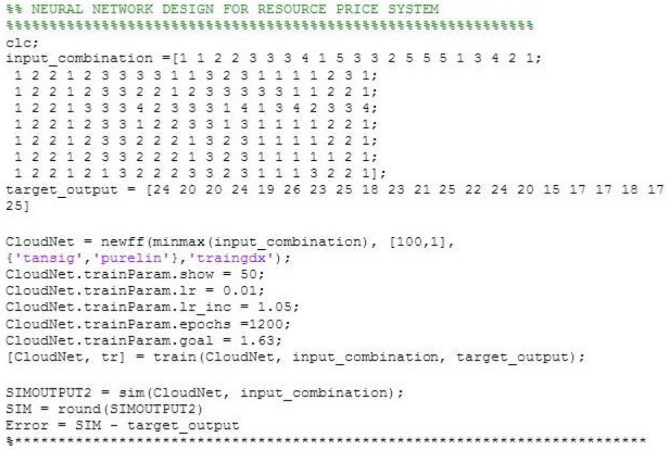
```
CloudNet = newff(minmax(input_combination),
    [100,1],
    {’tansig’,’purelin’},’traingdx’);
CloudNet.trainParam.show = 50;
CloudNet.trainParam.lr = 0.01;
CloudNet.trainParam.lr_inc = 1.05;
CloudNet.trainParam.epochs =1200;
CloudNet.trainParam.goal = 1.63;
[CloudNet, tr] = train(CloudNet,
    input_combination,
    target_output);
```



PR	MS	CR	SLA	UPL	UBR	UCR	SC	Input Combination	Output (RP)
1-Excellent	1-Cloup Provider	1-Good	1- Gold	1-Good Physical Location	1-Good Bank Record	1-Good Police Record	1-New Company	1,1,1,1,1,1,1,1 1,2,2,2,2,2,2,2	24 20
2-Very Good	2-3rd Party	2-Moderate	2-Silver	2-Moderate Physical Location	2-Moderate Bank Record	2-Moderate Police Record	2-Middle Age Company	2,2,2,2,2,2,2,2 2,1,1,1,1,1,1,1 3,2,2,3,2,2,2,2	20 24 19
3-Good	3-Not Available	3-Poor	3-Platinum	3- Bad Physical Location	3-Poor Bank Record	3-Poor Police Record	3-Old Company	3,3,3,3,3,3,3,1 3,3,3,3,3,3,3,3	26 23
4-Poor			4-User Defined					4,3,2,4,1,2,2,2 1,3,2,2,2,2,2,2	25 18
5-Bad								5,1,1,3,2,2,2,2 3,1,2,3,3,1,1,3	23 21
								3,3,3,3,3,3,3,3 2,2,3,1,1,2,2,2	25 22
								5,3,3,4,3,3,3,3	24
								5,1,3,1,1,1,1,1	20

Fig. 5: Sample Data for the RP ANN System

The ANN was simulated using the command `sim(CloudNet, input_combination)` and the CloudNet being the network object and different input combination were tried. The network simulation was static because the sequence or timing of the inputs is not important. The training type is incremental because the effect of the input on the network is not the same. The RP ANN system was implemented with Artificial Neural Network toolbox of Matlab. The codes were deposited in an m-file. The codes are shown in Figure 6.



```
%% NEURAL NETWORK DESIGN FOR RESOURCE PRICE SYSTEM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
input_combination =[1 1 2 2 3 3 3 4 1 5 3 3 2 5 5 5 1 3 4 2 1;
 1 2 2 1 2 3 3 3 3 1 1 3 2 3 1 1 1 1 2 3 1;
 1 2 2 1 2 3 3 2 2 1 2 3 3 3 3 3 1 1 2 2 1;
 1 2 2 1 3 3 3 4 2 3 3 3 1 4 1 3 4 2 3 3 4;
 1 2 2 1 2 3 3 1 2 2 3 3 1 3 1 1 1 1 2 2 1;
 1 2 2 1 2 3 3 2 2 2 1 3 2 3 1 1 1 1 2 2 1;
 1 2 2 1 2 3 3 2 2 2 1 3 2 3 1 1 1 1 1 2 1;
 1 2 2 1 2 1 3 2 2 2 3 3 2 3 1 1 1 3 2 2 1];
target_output = [24 20 20 24 19 26 23 25 18 23 21 25 22 24 20 15 17 17 18 17
25]

CloudNet = newff(minmax(input_combination), [100,1],
{'tansig','purelin'},'traingdx');
CloudNet.trainParam.show = 50;
CloudNet.trainParam.lr = 0.01;
CloudNet.trainParam.lr_inc = 1.05;
CloudNet.trainParam.epochs =1200;
CloudNet.trainParam.goal = 1.63;
[CloudNet, tr] = train(CloudNet, input_combination, target_output);

SIMOUTPUT2 = sim(CloudNet, input_combination);
SIM = round(SIMOUTPUT2)
Error = SIM - target_output
%************************************************************************
```

Fig. 6: Neural Network Design for Resource Price System

V. Use Cases

To illustrate the applicability of our *Cloud Resource Price System* a simple use case is introduced. As a precondition the

Cloud user has to fill in a web form with all required data, like the fix and variable parameters. Some information depend on input by the cloud provider, other by the cloud customer. The following information has to be provided by the Cloud Provider:

- The state and the location of the data centers, to determine their individual fix costs. All fix costs will be handled by the providers price manager. Assume a data center in Germany based on the total cost of ownership of a VM (2 Core, 2G Mem, 100G storage) has a fixed price of 8.50€.
- The SLA templates for *BRONZE*, *SILVER*, *GOLD* and for each offered QoS selectable for the customer has to be factored by the price manager of the provider.
- The Cloud Provider Reputation (CR) is expressed based on the availability of Past Experience or Compensation Records. Suppose the provider has both a "Record of Success" and a "Record of Compensation", then its reputation is *Good* =>, which will result in no change in the price.

The following information has to be provided by the Cloud customer:

- If the customer wants to define individual QoS, the price for each QoS is determined using information of user, who predicts his usage of the resource. If heavy usage is predicted during working hours it will costs more, than during the night. A detailed discussion about this can be found in the paper "Cloud Utility Price Model" [8]. So for example a customer has chosen *GOLD* (HA, backup every day, restore time < 2h, etc.), this could result in an increase of the price by 20$.
- The customer has to check in the template the type of monitoring services that are available. Monitoring service can be available from the provider itself or from a third party or may not be available. If no monitoring is selected => then there is no change in the price.
- The user reputation is measured using the "Physical Address", the "Bank Details" and the "Police Record". The particular option selected will determine the price to give. A user with a bad police record is a high risk user, hence the user has to pay higher price than others. In our example we assume being *Good* for all three reputations attributes.

After that, all information is fed into the price model, building the final price for the cloud service. First, all fix cost will be calculated with the mathematical formula and other information like state and location will be evaluated by the if then rules. Next step is the determination of information like reputation or customer behavior where the neural networks will predict the estimated price.

The provider itself has a provider price manager, where he can add and change the fix and variable costs. All these informations are used by the price modeling module, which calculates the price for the cloud resources. The user also checks the review from the public on this provider. Though the review does not affect the price heavily, but it has an influence.

VI. Conclusion

The price of a Cloud resource is influenced by many factors. It has been shown how a variety of static and dynamic factors like hardware price, data center location, provider or user reputation could be used to adapt the price of a cloud resource. To keep up with the dynamicity of the factors influencing the price, an adaptable Cloud Resource Price Model has been developed. By using a combination of mathematical formula, IF-THEN rules and a Neural Network this has been realized. A simple use case was introduced to show how useful this approach can be.

Acknowledgment

The study was carried out in the Cloud Research Lab of HFU. The support of the TWAS_DFG Cooperation Visit is appreciated. The research is supported by the German Federal Ministry of Education and Research (BMBF) through the research grant number 03FH046PX2.

References

[1] B. Singh, M. Kaur, and C. Batra, "A review of new about cloud computing security," INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY, vol. 4, no. 2b1, 2013.

[2] P. Wayner, "Cloud review: 8 public cloud services put to the test from amazon to windows azure, iaas clouds differ widely in features, complexity, and speed." InfoWorld, March 2013. [Online]. Available: http://www.infoworld.com/d/cloud-computing/cloud-review-8-public-cloud-services-put-the-test-214403?source=fssr

[3] A. M. Odlyzko, "Internet pricing and the history of communications," Computer Networks, vol. 36, no. 5/6, pp. 493–517, 2001.

[4] P. C. Fishburn and A. M. Odlyzko, "Competitive pricing of information goods: Subscription pricing versus pay-per-use," Economic Theory, vol. 13, no. 2, pp. 447–470, 1999.

[5] K. Lai, "Markets are dead, long live markets," CoRR, vol. abs/cs/0502027, 2005.

[6] A. Strømmen-Bakhtiar and A. R. Razavi, "Cloud Computing Business Models Cloud Computing for Enterprise Architectures," ser. Computer Communications and Networks, Z. Mahmood and R. Hill, Eds. London: Springer London, 2011, ch. 3, pp. 43–60. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-2236-4_3

[7] R. I. Gopalkrishnan, A. D. Miyazaki, D. Grewal, and M. Giordano, "Linking web-based segmentation to pricing tactics," Journal of Product & Brand Management, vol. 11, no. 5, pp. 288– 302, 2002.

[8] S. A. Bello and C. Reich, "Cloud utility price models." in CLOSER, F. Desprez, D. Ferguson, E. Hadar, F. Leymann, M. Jarke, and M. Helfert, Eds. SciTePress, 2013, pp. 317–320.

[9] D. Patel Chandrakant and J. Shah Amip, "Cost model for planning, development and operation of a data center," Internet Systems and Storage Laboratory, HP Laboratories, Palo Alto, Tech. Rep., 2005, hPL-2005-107(R.1).

[10] S. Maxwell, The Price Is Wrong: Understanding What Makes A Price Seem Fair And The True Cost Of Unfair Pricing. Hoboken, N.J.: John Wiley & Sons, 2008.

[11] D. Osterwalder, Trust Through Evaluation and Certification? Sage, 2001.

[12] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science, ser. CLOUDCOM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 693–702.

[13] 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, no. 4, pp. 50–58, Apr. 2010.

[14] P. Hofmann and D. Woods, “Cloud computing: The limits of public clouds for business applications.” IEEE Internet Computing, vol. 14, no. 6, pp. 90–93, 2010.

[15] S. Frey, C. Lüthje, R. Teckelmann, and C. Reich, “Adaptable Service Level Objective Agreement (A-SLO-A) for Cloud Services.” in CLOSER, F. Desprez, D. Ferguson, E. Hadar, F. Leymann, M. Jarke, and M. Helfert, Eds. SciTePress, 2013, pp. 457–462.

[16] Y. Chen, V. Paxson, and R. Katz, “What’s New About Cloud Computing Security,” University of California, Berkeley Report No. UCB/EECS-2010-5 January, vol. 20, no. 2010, pp. 2010–5, 2010.

[17] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, “On technical security issues in cloud computing,” in Proceedings of the 2009 IEEE International Conference on Cloud Computing, ser. CLOUD ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 109–116.

[18] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, “Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds,” in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS ’09. New York, NY, USA: ACM, 2009, pp. 199–212.

An Evolutionary Training Algorithm for Artificial Neural Networks with Dynamic Offspring Spread and Implicit Gradient Information

Martin Ruppert
Institute of Computer Science
Worms University of Applied Sciences
Worms, Germany
ruppert@fh-worms.de

Eric MSP Veith, Bernd Steinbach
Institute of Computer Science
Freiberg University of Mining and Technology
Freiberg, Germany
{veith, steinb}@informatik.tu-freiberg.de

***Abstract*—Evolutionary training methods for Artificial Neural Networks can escape local minima. Thus, they are useful to train recurrent neural networks for short-term weather forecasting. However, these algorithms are not guaranteed to converge fast or even converge at all due to their stochastic nature. In this paper, we present an algorithm that uses implicit gradient information and is able to train existing individuals in order to create a dynamic reproduction probability density. It allows us to train and re-train an Artificial Neural Network supervised to forecast weather conditions.**

Keywords–artificial neural network; evolutionary algorithm; weather forecasting; smart grid.

I. Introduction

Machine learning finds its application in many areas. One of them is short-term weather forecasting, which is useful for predicting the output of renewable energy sources [1]. The basic assumption that wind speed or solar radiation follow a particular, detectable pattern introduces Artificial Neural Networks (ANN) as a probable device for forecasting. The simplest form of the ANN, the Perceptron, is primarily usable for detecting static patterns.

However, the more variety input data has, the larger the error of a Perceptron. Introducing ANNs with a short-term memory that implement the concept of time, such as Elman's [2], increase the success of the ANN.

Traditional training methods based on backpropagation, such as RPROP [3] and its variants, allow fast weight updates in online or stochastic mode, i.e., immediately after a pattern has been seen. However, these algorithms can get stuck in local minima. Evolutionary algorithms [4] can solve this problem by introducing randomness through their process of mutation and crossover, which also includes seemingly bad individuals. This offers a chance to escape a local minimum. However, this randomness typically increases the time it takes for an evolutionary algorithm to arrive at a properly trained ANN, and it can even be unsuccessful.

In this paper, we present a training method that combines the gradient descent technique of a backpropagation-based training method with the resilience of an evolutionary algorithm against local minima.

II. Motivation

Since the search space during the training for artificial neural networks is big for any real-life application, many training functions harbor the danger of getting stuck in local minima. Evolutionary training methods circumvent this by introducing randomness into the process. However, this, in turn, increases the training time and does not guarantee success per se. Results obtained by training using evolutionary algorithms can even yield worse results, since, by purpose, there is no knowledge of an error gradient included.

This problem becomes apparent when using artificial neural networks for weather forecasting since the search space of a wind profile offers many local minima.

In [5], Maqsood *et al.* use an ensemble of neural networks to forecast weather. Although this ensemble technique is successful, it requires four different networks in order to yield these results. Moreover, the networks are retrained for the four seasons (winter, spring, summer and fall) separately. For this supervised training, they use a hand-selected sample set.

If forecasters are installed at different sites, supervised training will have to occur separately for each node, because their different locations mean different weather conditions. The ANN will have to continuously adapt itself in order to remain reliable even under uncommon weather conditions. Since remote sites such as wind parks will typically feature embedded systems, training should occur in a short period of time. However, using backpropagation-based algorithms will yield non-optimal results due to the algorithm getting stuck in local minima, which will introduce large deviation in cases of, e.g., gusts of wind.

To address this problem, we propose a combination of both evolutionary training and deterministic training algorithms that can use the advantages of both approaches. Our approach, which employs evolutionary strategies, uses information about the current success and implicit gradient information when creating the offspring. Furthermore, we allow existing individuals to be improved instead of resorting to improve the overall population through the offspring only.

The remainder of this paper is structured as follows. We describe the training algorithm in the Section III along with a pseudo-code representation in Figure 2. We discuss our approach in Section IV and conclude in Section V.

III. The Training Algorithm

The algorithm currently finds its application in training neural networks that follow the design of Elman [2]. The difference to Elman's design lie in the connections to the context layer: The hidden layer is fully connected to the

ISBN: 978-1-61208-357-5

context layer. Furthermore, these connections can be trained, i.e., their weight is not set fixed to 1.0.

To the algorithm, the concepts of “neural networks” with “trainable weights” do not exist. Instead, we operate on individuals that we call *objects*. These objects, in turn, that have *parameters*. This application-agnostic approach is consistent with literature. It will, in the future, also allow us to apply the algorithm to other problems instead of constraining it to artificial neural networks. For ANNs, a parameter is a particular, trainable weight.

Additionally to their parameters vector, each object also has a *scatter* vector, **s**. It vector limits the interval of modification during an iteration, t, for each parameter p_i:

$$p_{t,i} = [-s_i \cdot p_{t-1,i}, s_i \cdot p_{t-1,i}] \tag{1}$$

As soon as an object is evaluated, its fitness is stored in its *fitness* vector. The total mean error is stored in f_0, while the mean error values of different samples are stored in $f_1, \ldots, f_n$.

Each object has additionally a maximum *age* that limits the number of iterations it may exist.

Since many steps of the training process require random numbers, we define a function called *frandom()* that returns a random number in the interval $[0.0, 1.0)$ with an uniform distribution. We can create different points of high density of the uniform distribution by calling *frandom()* multiple times. The calls are concatenated by addition or subtraction, depending on where we want these points to be. This uniform distribution is used to pick initial values for scatter and parameters for all objects derived from the user-supplied base object during the creation of the initial population.

We begin by creating the starting population. Each population consists of a number of objects that are active, i.e., trainable. This upper bound of the size of a population is contained in the variable *numActiveObjects*.

The training algorithm has the notion of an elite, i.e., a number of objects that are considered to be better than the rest of the population. The elite is included in the maximum number of objects in the population, i.e., $population = elite \cap others$. The size of the elite never changes: This is a user-configurable value. However, as soon as any other object outside the elite is better than any elite object, it is exchanged with the worst elite object.

The initial set of parameters is supplied by the user. The initial scatter vector is filled with random numbers within bounds supplied by the user. However, only the first, i.e., the *base object* o_0, of the initial population gets these pristine values assigned; for all other generated objects (o_n) these are modified according to Equations 2 and 3.

$$o_n.s_i = o_0.s_i \cdot \exp(4.0 \cdot (0.5 - frandom())) \tag{2}$$

$$o_n.p_i = o_0.p_i \cdot s_i - \sum_0^3 frandom() \tag{3}$$

The user also supplies the fitness function, *fitness(o)*. It is required to return the fitness value of the object, o, as a single float-point number in order to compare it to the user-supplied target fitness. The fitness value is also used to sort the population. Since our training strategy is application-agnostic, it does not evaluate the ANN directly; this is the user's responsibility.

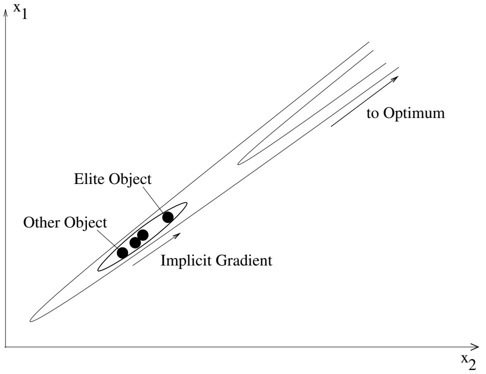


Figure 1. The implicit gradient information used by the modified reproduction function, *GenerateObject()*

To finalize the initialization phase, the population's fitness values are determined and the population is sorted accordingly.

The population is then iteratively improved until either the maximum number of iterations is reached (*maxIter*), there has been no improvement for a designated number of iterations (*maxNoSuccess*) or the user's target fitness (*targetFitness*) value has been reached.

A global increase in fitness means that a new best object (o_b) has been found. This also sets *lastSuccess* to the current iteration in order to make this fact known to the outer loop. Otherwise, it would break on $iter - lastSuccess >= maxNoSuccess$. The error of the new best object is then, at the end of the iteration, compared to the user's target fitness. If it is equal or better, the training ends and the newly found best object is returned.

The training function takes samples of the success of the training at a constant interval, T. This is used to calculate the mean success dynamically by using a Linear Time-Invariant system (LTI). This LTI is defined as:

$$pt1(y,u,t) = \begin{cases} u & \text{if } t = 0 \\ y + \dfrac{u-y}{t} & \text{otherwise.} \end{cases} \tag{4}$$

with $t = T$ in our case. The constant T is user-defined and denotes the number of iterations between two samples. The *pt1(y, u, t)* function is called after a newly generated object has been tested for its fitness. If it is better than the worst object of the current population, we set $success = pt1(success, 1.0, T)$. But, only if this worst object has still iterations to live; if not, we set $u = -1.0$ in the call to *pt1(y, u, t)*, since replacing an already dead object cannot be counted as success.

This mean success is important during the generation of new objects, because it is used in order to calculate the *implicit gradient information*. These information are used to calculate a new object's parameters. Figure 1 shows schematically how a set of objects uses the implicit gradient information to move towards the optimum.

```
global $population, eliteSize, success, targetSuccess, successWeight, gradientWeight$
local $o_n, o_e, o_r, i_1, i_2, gradientSwitch, \Delta x, succcessRate, expvar, xlp$
$xlp \leftarrow 0.0$
$i_1 \leftarrow \mid \text{RANDOM}() \mod eliteSize - \text{RANDOM}() \mod eliteSize \mid$
$i_2 \leftarrow \text{RANDOM}() \mod population.length$
if $\text{OBJECT1ISBETTER}(population_{i_2}, population_{i_1})$ then $\text{SWAP}(i_1, i_2)$
$o_e \leftarrow population_{i_1}$
$o_r \leftarrow population_{i_2}$
$successRate \leftarrow success/targetSuccess - 1.0$
$gradientSwitch \leftarrow \text{RANDOM}() \mod 3$
if $gradientSwitch = 2$
  then $xlp \leftarrow (\sum_0^9 \text{FRANDOM}() - \sum_0^5 \text{FRANDOM}()) \cdot gradientWeight$
       if $xlp > 0.0$ then $xlp \leftarrow xlp \cdot 0.5$
       $xlp \leftarrow xlp \cdot \exp(gradientWeight \cdot successRate)$
$expvar \leftarrow \exp(\text{FRANDOM}() - \text{FRANDOM}())$
for $i \leftarrow 0$ to $o_n.\mathbf{p}.length$
  do $\Delta x \leftarrow o_e.s_i \cdot \exp(successWeight \cdot successRate)$
     $o_e.s_i \leftarrow \text{APPLYBOUNDSFROMEQUATION } 6(\Delta x)$
     if $\text{FRANDOM}() < 0.5$ then $\Delta x \leftarrow o_e.s_i$ else $\Delta x \leftarrow 0.5 \cdot (o_e.s_i + o_e.s_i)$
     $\Delta x \leftarrow \Delta x \cdot expvar$
     $o_n.s_i \leftarrow \text{APPLYBOUNDSFROMEQUATION } 6(\Delta x)$
     $\Delta x \leftarrow o_n.s_i \cdot (\sum_0^4 \text{FRANDOM}() - \sum_0^4 \text{FRANDOM}())$
     if $gradientSwitch = 0$
       then {if $\text{RANDOM}() \mod 3 < 2$ then $\Delta x \leftarrow \Delta x + o_e.p_i$ else $\Delta x \leftarrow \Delta x + o_r.p_i$
       else if $gradientSwitch = 1$ then $\Delta x \leftarrow o_e.p_i$
       else if $gradientSwitch = 2$
       then $\Delta x \leftarrow \Delta x + o_e.p_i$
            $\Delta x \leftarrow \Delta x + xlp \cdot (o_e.p_i - o_r.p_i)$
     $o_n.p_i \leftarrow \Delta x$
return $(o_n)$
```

Figure 2. The GENERATEOBJECT() function

In *GenerateObject()* as detailed in Figure 2, we pick a random elite object (o_e) and a random object of the whole population (o_r) in order to create the new offspring, o_n.

For this, we determine the influence of the implicit gradient information, *xlp*. It is used on a random basis with a probability of $p = \frac{1}{3}$. This prevents *GenerateObject()* from completely discarding objects with bad gradients. Discarding only happens because of an object's age. If it is used, we first create a custom uniform distribution by repeated calls to *frandom()*. We further modify *xlp* by $\exp(successRate \cdot gradientWeight)$.

The user is able to tune the influence of the implicit gradient information by modifying the Variable *gradientWeight*. Our experiments have shown that values in the range of $[1.0, 3.0]$ show great success. A value of 0.0 completely disables this feature. Similarly, the influence of the mean success can be disabled by setting *successWeight* to 0.0.

The actual delta (henceforth Δx) by which first the object's scatter and then its parameters are modified is first derived from the elite object's scatter as shown in Equation 5.

$$\Delta x = \begin{cases} 0.5 \cdot (o_e.s_i + o_r.s_i) & \text{if } frandom() < 0.5 \\ o_e.s_i \cdot \exp(successWeight \cdot success) & \text{otherwise.} \end{cases} \tag{5}$$

The bounds specified in Equation 6 are then enforced.

$$eamin \leq ebmin \cdot \mid o_e.p_i \mid \leq \Delta x \leq ebmax \cdot \mid o_e.p_i \mid \tag{6}$$

The three variables the define the limits have the following meanings: *eamin* is the absolute minimum for values and typically set to the smallest IEEE 32 Bit floating point number, i.e., $1 \cdot 10^{-32}$. *ebmin* is the relative minimum of scatter. It is user-tunable, but $(ebmin + 1.0) > 1.0$ must be true. *ebmax* is the relative maximum of scatter. It is also user-tunable. We suggest $ebmin < ebmax < 10.0$ per the results of our experiments.

The scatter is finally used to set the new object's parameters.

We detail the complete process of generating a new object in Figure 2.

IV. DISCUSSION

The algorithm's two primary advantages over traditional evolutionary algorithms are its ability to use implicit gradient information and the dynamic density by which new objects spread out in the search space.

We call this dynamic attribute of the algorithm *reproduction probability density function*. It is controlled by the relationship of the two variables *success* and *targetSuccess*. If *success* > *targetSuccess*, the spread of new objects is increased. It is decreased if the opposite holds true.

The function becomes obvious in *GenerateObject()*. Here, the current success rate influences not only the new object's scatter and parameter vector, but also those of the selected elite object. This way, objects move dynamically towards a minimum in the search space. Thus, the training algorithm does not only work by iteratively creating new object through mating and crossover, but also enables older elite objects to "learn" and improve.

An additional piece of information we can draw from the success of the different objects is an implicit gradient information. Implicitly, because it is available through the spread of the selected objects towards a minimum. It is most obviously in the assignment in Equation 7.

$$\Delta x = \Delta x + xlp \cdot (o_e.p_i - o_r.p_i) \qquad (7)$$

However, using these information also harbors the danger of converging towards a local minimum instead of a global one. An evolutionary algorithm typically saves the user from this by its random crossover and mutation procedures which always carry a chance of escaping a local minimum. We also include this behavior since we enable this feature on a random basis via the *gradientSwitch* variable.

Preliminary tests have been conducted using 10 minutes mean wind speeds obtained from Germany's national weather service, DWD. We have compared the testing performance of our algorithm to that of Simulated Annealing [6]. In order to make the results comparable, both implementations use the same code base.

For the comparison, two independent ANNs have been identically configured and initialized. Both algorithms continuously trained their neural network with the same data. For each forecast, the ANNs have been fed with the last 12 10 minutes mean values in order to make use of the short-term memory the Elman ANN provides. The network was then used to forecast the next 10 minutes. The network's forecast was finally compared to the actual measurement provided by the national weather service in order to calculate the network's error.

During the training phase, our algorithm showed an almost constant training time with a variation of $\Delta t \leq 1s$. This substantiates that the population had a nearly identical "way to travel" to an optimum during re-training, doing so targeted based on the implicit gradient information. The Simulated Annealing algorithm, in contrast, produced widely varying training times. On average, our algorithm needed 5% of the time Simulated Annealing took.

In the day period of which Figure 3 shows a section, the mean error of the ANN our algorithm trained was 1.31, while the network the Simulated Annealing algorithm trained obtained a mean error value of 1.86.

Figure 3 depicts a representative section of a test run. One can observe the varying training time of the Simulated Annealing algorithm due to its completely stochastic nature, while our algorithm shows constant training time. The three large spikes in the error values come from turbulences measured; interestingly, the network trained with our algorithm was able to forecast two of the three.

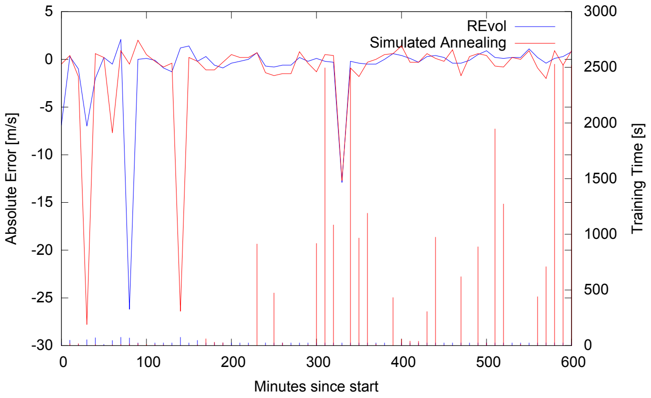


Figure 3. Absolute error and training duration of our algorithm ("REvol") and Simulated Annealing

V. Conclusion and Future Work

In this paper, we presented a training algorithm for ANNs that combines the strengths of evolutionary algorithms and deterministic ones. Our algorithm is able to include implicit gradient information into the reproduction process and allows training of already existing objects. It reduces the possibility of getting stuck in a local minimum, because it is based on paradigm of evolutionary algorithms that introduce randomness in order to escape local minima.

Due to these two features, we expect our algorithm to converge on a good minimum with a higher probability while still being able to escape a local minimum.

In the future, we will test our approach against other algorithms in terms of speed and convergence towards good minima. We will especially pay attention to Long-Short Term Memory approaches.

VI. Acknowledgments

This paper has been created as part of a cooperative doctorate program between the TU Bergakademie Freiberg and Wilhelm Büchner Hochschule, Pfungstadt.

References

[1] C. Potter, A. Archambault, and K. Westrick, "Building a smarter smart grid through better renewable energy information," in Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES, March 2009, pp. 1–5.

[2] J. Elman, "Finding structure in time," Cognitive Science, vol. 14, no. 2, Jun. 1990, pp. 179–211.

[3] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," in IEEE International Conference on Neural Networks, 1993, pp. 586–591.

[4] J. Branke, "Evolutionary algorithms for neural network design and training," in Proceedings of the First Nordic Workship on Genetic Algorithms and its Applications, 1995, pp. 145–163.

[5] I. Maqsood, M. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," Neural Computing and Applications, vol. 13, no. 2, May 2004, pp. 112–122.

[6] S. Russel and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed. Prentice Hall, 2010.

Energy Consumption Optimization through Pre-scheduled Opportunistic Offloading in Wireless Devices

Constandinos X. Mavromoustakis
Department of Computer Science
University of Nicosia
46 Makedonitissas Avenue, P.O.Box 24005,
1700 Nicosia, Cyprus
mavromoustakis.c@unic.ac.cy

Andreas Andreou
Faculty of Computer Science and Technology
University of Cambridge
William Gates Building, 15 JJ Thomson Ave,
Cambridge CB3 0FD, UK
aa773@cam.ac.uk

George Mastorakis, Stelios Papadakis
Department of Business Administration
Technological Educational Institute of Crete
Lakonia, Agios Nikolaos, 72100, Crete, Greece
gmastorakis@staff.teicrete.gr, spap@staff.teicrete.gr

Athina Bourdena, Dimitris Stratakis
Department of Informatics Engineering
Technological Educational Institute of Crete
Estavromenos, Heraklion, 71500, Crete, Greece
bourdena@pasiphae.eu, dstrat@ie.teicrete.gr

***Abstract*—The current research on mobile cloud computing systems and mechanisms has identified several challenges that have to be addressed for permitting execution on remote terminals/servers. A mobile cloud computing service provision has to be based on a framework that will ensure the effective execution of applications under an energy-efficient approach. In this context, this paper elaborates on the assessment of a framework that exploits a cooperative process-execution offloading scheme, pointing at offering energy conservation. The proposed approach utilizes a dynamic scheduling process to ensure that no discontinuous execution will happen on mobile devices. In addition, this paper elaborates on a partial offloading algorithm for an energy-efficient failure-aware allocation of the resources, by considering temporal execution-oriented metrics for the evaluation of the performance. The proposed scheme is analytically assessed through event driven simulation tests, towards verifying the effectiveness of the anticipated offloading approach, in terms of the energy consumption of the mobile devices and the quality of the degree offered.**

Keywords- Mobile cloud; offloading methodology; temporal execution-oriented metrics; opportunistic cloud reliability; dynamic resource migration.

I. Introduction

The increasing number of the mobile devices (e.g. smart phones, tablets) introduces a new ground of research in the field of mobile wireless communications. This development has additionally been powered by a large number of applications that can be exploited in each mobile device, making the need for a dependable and high execution mobile computing processing environment. The proposed approach adopts an offloading process to partially outsource the resources that are required on a server rack or on an alternative mobile device with redundant resources. This offloading mechanism is adopted as a part of the application start-up, towards minimizing the GPU/CPU efforts, as well as the energy that is consumed by the mobile device, running out of resources.

Mobile cloud computing services have to be provided according to a synchronous mode [1], while several metrics in terms of the mobile devices, as well as the availability of the offloading by other terminals or serves have to be also taken into account [1]. The current cloud computing models are considered as 'low offered' throughput models [2], [3], while sometimes they offer low Quality of Service (QoS) or Quality of Experience (QoE) to the end-recipients (i.e., mobile devices users). The restrictions regarding the processing power, as well as the bounded capacity availability of mobile devices, aggravate the execution and harmfully affect the consistency offered to the mobile user, due to capacity-oriented failures and intermittent execution. If the available resources (processing and/or memory-oriented) are restricted, the mobile devices could exploit a cloud network infrastructure [1], towards supporting precise execution, via a resource/task migration mechanism. Such a mechanism has not yet been investigated to guarantee that satisfactory processing power is available, towards executing the application of a mobile device.

In this framework, this paper elaborates on a dynamic scheduling scheme, towards enabling offloading of the resources from a mobile device to another mobile device. The proposed scheme reduces the resources utilization of the mobile devices (GPU, CPU, RAM, battery consumption), by supporting at the same time, the extensibility of their lifetimes. The proposed approach in this paper is adopted to minimize the computational load of mobile devices, towards extending the lifetime of their battery. In addition, this approach takes into account a partitionable parallel processing wireless system, where the resources are partitioned and handled by a subsystem [1], evaluating the resource offloading process. Finally, a certain algorithm is proposed for the offloading process, towards dynamically defining the optimal resource manipulation in an energy-efficient approach.

In this context, the sections of this paper are based on the following structure. The related work and the research motivation are described in Section II. This section particularly focuses on the current research approaches, as well as on the resource offloading/migration scheduling policies. Section III then elaborates on the proposed offloading scheme and the associated mechanisms to reduce the energy consumption, maximizing the lifetime of the mobile devices. The proposed approach is based on the available resources of the mobile devices, the temporal terminals characteristics and the server-based parameters, along with the communication-oriented diversities. This approach establishes and maintains an efficient resources manipulation in the mobile devices, under an energy-efficient manner. Section IV presents the results that were obtained, by conducting simulation experiments, towards evaluating the performance of the proposed mechanism, by focusing on the behavioral characteristics of the scheme, along with the system response, as well as the energy consumption achieved. Finally, Section V summarizes the research findings of this paper and discusses the potential future directions for further experimentation and research.

II. RELATED WORK AND RESEARCH MOTIVATION

It is definitely valid that over the last years, a number of research efforts have been dedicated to device-to-device or Machine-to-Machine communication networks, ranging from physical layer communications to communication-level networking challenges. Since mobile devices are able to exchange resources through mobile networks, they generate large amounts of data [2]. The QoS [3] offered by these devices where on-the-move services are taking place is aggravated significantly by energy-hungry applications (e.g., video services, interactive gaming, etc.). In this context, the explicit lifetime of such devices has to be extended, especially when energy-hungry applications are exploited. In addition, efficient resource management has to be achieved within the context of the cloud computing paradigm and effective allocation issues of the processor power, the memory capacity resources, as well as the network bandwidth have to be considered. The resource management mechanisms have to allocate the resources of the mobile devices, on a cloud-based infrastructure, towards migrating some of their resources on the cloud servers [4]. The mobile devices have also to operate under specific QoS requirements, as defined by the users and the applications characteristics. The resource management process at the cloud scale involves a rich set of resource and task management schemes, which are able to effectively manage the QoS provisioning, by preserving the efficiency of the total system. However, the energy efficiency issue is one of the greatest challenge for this optimization problem, along with the offered scalability in the framework of the performance evaluation and measurement. In this context, different dynamic resource allocation policies have been explored in [5], towards elaborating on the enhancement of the application execution performance and the efficient utilization of the resources. Other research approaches associated with the performance of dynamic resource allocation policies, had led to the development of a computing framework [6] that takes into account the countable and measureable parameters affecting the task allocation.

In addition, authors in [7] address this problem, by using the CloneCloud approach [8] of a smart and efficient architecture. This architecture is exploited for the seamless use of the ambient computation to augment mobile device applications, off-load the right portion of their execution onto device clones and operate in a computational cloud. Authors in [8] statically partition service tasks and resources between the client and the server portions. The service is then reassembled on the mobile device at a later stage. The research approach in [8] is based on a cloud-augmented execution, by exploiting a cloned VM image as a powerful virtual device. This approach has many weaknesses, since it considers the resources of each cloud rack, depending on the expected workload and execution conditions (CPU speed, network performance). In addition, a computation offloading scheme is proposed in [9] that is exploited in cloud computing infrastructures to minimize the energy consumption of a mobile device, enabling the execution of certain/specified and under constrains applications. Energy consumption issues have also been investigated in [10], towards supporting computation offloading in a combination of 3G and Wi-Fi network infrastructures. However, such evaluations do not maximize the benefits of offloading, as they are considered as high latency offloading processes and require low amount of information to be offloaded. Cloud computing is currently impaired by the latency that is experienced during the data offloading through the wireless network infrastructure. Towards this direction, authors in [10] and [1], elaborate on issues, where the mobile devices exploit delay sensitive services. However, the variability of this delay in turn impairs the QoS/QoE of the mobile users.

Furthermore, authors in [11] elaborate on the resource processing poverty for ‘energy-hungry’ applications that request large amount of processing resources to run on a mobile device. Authors in [12] propose a resource manipulation scheme as a solution that is based on the failure rates of cloud servers in a large-scales datacenters. However, such criteria do not include the communications diversities of the servers during the communication process with the mobile users’ claims. This approach also does not take into account the available processing resources, the utilization of the device memory, the remaining energy and the available capacity with the communication of each of the device with the closest –in terms of latency- cloud terminal.

Within this context, this paper proposes an offloading resource mechanism, which is used in collaboration with an energy-efficient model. The scheme exploits an offloading methodology in order to guarantee that no intermittent execution will occur on mobile devices, whereas the application explicit runtime will meet the required deadlines to fulfil the QoS requirements. This paper also elaborates on the development of an offloading scenario, in which the scheduling policy for guaranteeing the efficiency in the execution of mobile users' tasks/applications can be achieved in an energy-efficient manner. The proposed framework is thoroughly evaluated through event driven simulation experiments, in order to define the efficiency of the proposed offloading policy, in contrast to the energy consumption of the wireless devices, as well as for the reliability degree offered.

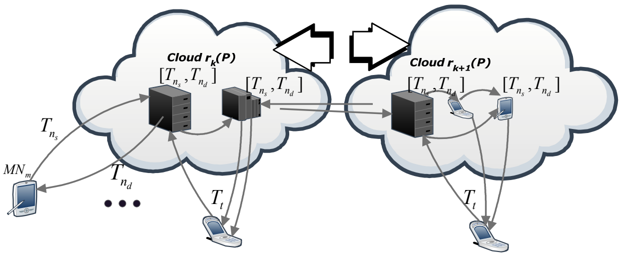


Figure 1. Cloud configuration and offloading process in order to achieve the best effort processing on-device power.

III. PRE-SCHEDULED OPPORTUNISTIC OFFLOADING IN WIRELESS DEVICES

A. Pre-scheduled offloading mechanism

Due to the heterogeneity in the hardware of both mobile devices and the servers on the cloud that the resources will be potentially (based on the proposed scheme) offloaded, the proposed framework encompasses the execution environment volatility and considers the cloud servers' response time, in order to a-priori compare them and select the appropriate server, according to the best fit-case. More specifically, this work considers the network-oriented parameters for bandwidth provisioning to achieve an acceptable resource offloading downtime δ (e.g., $\delta \le 1.6\,s$ as the experimental process validates in [1]). To this end, from the network perspective, the modelled parameters can be expressed, for an offloading process *O for an* executable resource task O_{a_j} , as a 5-tuple given by:

$O_{a_j}(MN)$ = <n_s, T_{n_s} , T_{n_d} , BW, T_t >, for the a_j executable task, where n_s is the devices or cloud terminals that the a_j from MN device will be offloaded, T_{n_s} is the source location best effort access time, T_{n_d} is the destination device or cloud location best effort access time from the source, BW is the required connection bandwidth and T_t is the connection holding duration for the a_j executable resource task.

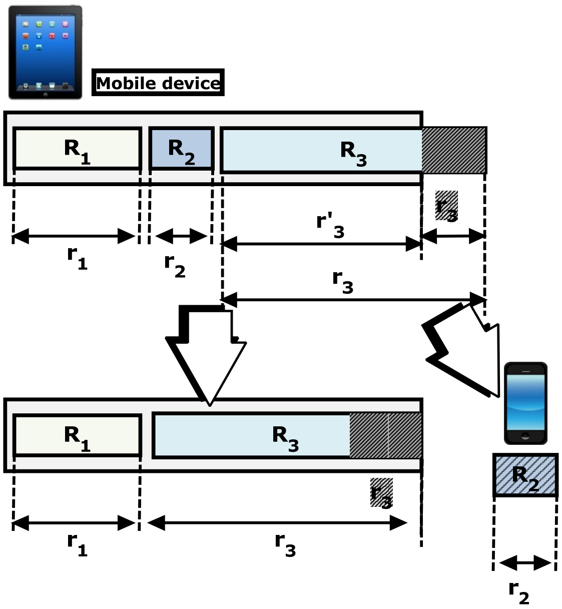


Figure 2. Resource partitioning onto mobile device.

In essence, the work done in [1] considers the resource transfer time, by taking into account the volume of traffic that should be transferred to the destination-source node. The total data volume that will be transferred, if the request meets the BW criteria can be provided by BW × T_t. In this work, the typical values ranges that were utilized in our experimental processes were $1MBps \le BW \le 20MBps$ and T_t=2s+t_x, $s = T_{n_s} + T_{n_d}$, where t_x is the time to process x-partitionable parts that are processed during the offloading process. Every executable resource task may have *x-partitions*, which in this work are considered as t_x partitioning parts/tasks where 1 ≤ t_x ≤ z**P,* where *z* is the number of different devices that the resource can be offloaded. Therefore, the number of tasks per executable resource task is limited to the number of terminals in the system. An executable resource can be shared and partitioned to x_1, x_2, ... x_n, while it can be simultaneously processed with r sequential partitions, where 0 ≤ r < *z*P*, if and only if the following relation holds:

$$r + \sum_{i=1}^{n} p(x_i) \le z*P \qquad (1)$$

where $p(x)$ represents the number of cloud terminals (mobile and statically located) that are needed to host the a_j. The scheduling strategy that was used is based on the Largest First Served/ Shortest Sequential Resource First Served and the service notations of [1] with a-priori knowledge of the $[T_{n_s}, T_{n_d}]$ service durations, as shown in Fig. 1. Every device with executable resource tasks and limited resources (memory) to execute, has to consider the offloading mechanism. In addition, Fig. 2 shows the dynamic offloading scheme, by considering executable resource tasks (t_x partitions) that will be offloaded according to the scheme proposed in the next section to conserve energy. In this scheme, partitionable tasks are offloaded onto other devices or cloud terminals based on the evaluation mechanisms shown in the next section, aiming at conserving energy on each mobile device, while running energy draining applications.

B. *Energy-consumption model using temporal capacity measurements*

Wireless nodes are error-prone with limited battery power, vulnerable and uncertain reliability, hosting energy-hungry applications. Therefore a challenging aspect for these devices is to design a resource offloading scheme that will significantly minimize the energy consumption and at the same time will enable the reliability in the smooth execution of any resource. As the consumed power varies with traffic and depends on the variations of the signal characteristics, as well as on the traffic-aware measurements [13], it is desirable to minimize the amount of power consumption, according to the resources that cannot 'run' on the mobile device or devices. To this end, the proposed scheme in this paper makes a progress beyond the current state-of-the-art, by elaborating on the association of the measurements of the partitionable tasks for two distinct cases: when resources can run on the devices. In this case, towards achieving energy conservation, the resources may be offloaded to a cloud or any other peer-neighboring device (so that the device that needs to run may potentially conserve energy); and the case that the device or devices cannot run the resources (as in Fig. 2) as the processing and memory requirements cannot support this execution. Thus, the measurable energy consumption can be evaluated according to the:

$$E_{r(a_j)} = E_c(a_j) \cdot \frac{C}{S_{a_j}} \tag{2}$$

In (2), $E_{r(a_j)}$ is the energy consumption, the parameter C is the number of instructions that can be processed within T_t, parameter S_{a_j} is the processing time of the server and $E_c(a_j)$ is the relative energy consumption expressed as:

$$E_c(r_i) = \frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c \tag{3}$$

where S_c is the server processing instruction speed for the computation resources, $Cost_c$ the resources processing instruction cost for the computation resources W_c energy consumption of the device or server in mW.

Each mobile device examines, if all neighboring 2-hops devices (via lookup table) can provide information about their offloading capabilities without affecting their energy status (thus without draining their energy to run other devices resources). In addition, the closest cloud rack is considered, if the relations exposed in (4) and (5) are not satisfied. Hence, for the neighboring devices *N* within *2-hops vicinity coverage* (based on the maximum signal strength and data rate model [1]) should stand:

$$\frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c \Big|^{r_i} > \frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c \Big|^{1,2..N} \tag{4}$$

$$W_{r_i} > W_c \forall 1,2,3,...N \text{ devices} \tag{5}$$

The energy consumption of each device should satisfy the (4)-(5) for each of the resources (executable processes) running onto the device MN_{m-1} hosting the r_i. Otherwise, the r_i with the maximum energy consumption is running in a partitionable manner to minimize the energy consumed by other peer-devices. These actions are shown in the steps of the proposed algorithm in table I.

TABLE I. DYNAMIC RESOURCE-BASED OFFLOADING SCHEME

```
Inputs: $MN_m$, Location($[T_{n_s}, T_{n_d}]$), resources
  $r_{1,} r_2, r_3, ..r_i \forall MN_m$
for all Cloud devices that stands $r_{1,} r_2, r_3, ..r_i$ find the $r_i$
  that can be offloaded to run onto another device
for all $MN_{m-1}$ do{
Estimate the $N_i$ //(as derived in ( 4))
  if ( $N_i$ is above a threshold){
    while ($T_t$== TRUE) {
      while ($1 \le t_x \le z*P$)
        search for $MN_{m-1}$ device that satisfies
          $\frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c \Big|^{r_i} > \frac{Cost_{c(r_i)}}{S_{c(r_i)}} \cdot W_c \Big|^{1,2..N}$, $W_{r_i} > W_c \forall 1,2,3,...N$
    offload ( $r_i, MN_{k(i)}$) //to $MN_{k(i)}$ to execute resource (i)
      end while
    end while ($C_{a_j} = \frac{T_k^j}{\sum_k T_{a_j}(r)} \forall \min(E_c(r_i)) \in 1,2,..N$)
   end for
  end if
end for
```

The resource allocation will take place, towards responding to the performance requirements as in [1]. A significant measure in the system is the availability of memory and the processing power of the mobile cloud devices, as well as the server-based terminals. The processing power metric is designed and used to measure the processing losses for the terminals that the r_i will be offloaded, as in (6), where a_j is an application and T_k^j is the number of terminals in forming the cloud (mobile and static) rack that are hosting application a_j and $T_{a_j}(r)$ is the number of mobile terminals hosting process of the application across all different cloud-terminals (racks).

$$C_{a_j} = \frac{T_k^j}{\sum_k T_{a_j}(r)} \forall \min(E_c(r_i)) \in 1, 2, ..N \qquad (6)$$

The Eq. (6) shows that if there is minimal loss in the capacity utilization i.e., $C_{a_j} \cong 1$ then the sequence of racks $T_{a_j}(r)$ are optimally utilized. The latter is shown through the conducted simulation experiments in the next section. The dynamic resource migration algorithm is shown in Table I with the basic steps for obtaining an efficient execution for a partitionable resource that cannot be handled by the existing cloud rack and therefore the migration policy is used to ensure that it will be continuing the execution. The continuation is based on the migrated policy of the partitionable processes that are split, in order to be handled by other cloud rack terminals and thus omit any potential failures. The entire scheme is shown in Table I, with all the primary steps for offloading the resources onto either MN_{m-1} neighbouring nodes or to server racks (as in [1]) based on the delay and resources temporal criteria.

IV. Performance Evaluation Analysis, Experimental Results and Discussion

The mobility model used in this work is based on the probabilistic Fraction Brownian Motion (FBM) adopted in [13], where nodes are moving, according to certain probabilities, location and time. Towards implementing such scenario, a common look-up application service for resource execution offloading is set onto each one of the mobile nodes MN_m. Topology of a 'grid' based network [1] was modeled, where each node can directly communicate with other nodes. For the simulation of the proposed scenario, the varying parameters described in the previous section were exploited, by using a two-dimensional network, consisting of nodes that vary between 10-150 (i.e., terminal mobile nodes) located in measured area, as well as five cloud terminals statically located on a rack. All measurements were performed using WLAN (Wi-Fi based technology specifications) varying with different 802.11X specifications. During simulation the transfer durations are pre-estimated or estimated, according to the relay path between the source (node to offload resources) and the destination (node to host the executable resources).

In this direction, Fig. 3 shows the number of requests (total over failed) with the number of mobile devices participating in the evaluated area. It is important to mark that when the dynamic offloading scheme takes place the total failed requests among nodes are significantly reduced, particularly when the nodes number is small. Towards examining the impact of the different capacities, several sets of experiments were conducted, using the presented resource offloading scheme. Large memory resources are executable resources/processes that are between 500 MBytes -1 GBytes, whereas small resources are executable resources/processes that host capacity between the range of 10-400 MBytes.

In addition, the partitionable task offloading on different mobile devices, on which the proposed offloading procedure takes place, is shown in Fig. 4, in contrast to the Throughput response of the system. Throughput response is presented with the number of the 'in-service' terminals on the cloud racks with different processing power and speed characteristics, as shown in Table II. Fig. 4 indicates that for large memory -required- resources and when the mobile terminals have no remaining memory to process these resources, throughput dramatically drops. Furthermore, the Service Time with the number of racks is shown in Fig. 5. The Service Time is greater for large files that are not migrated in partitionable parts to the other terminals on the cloud racks.

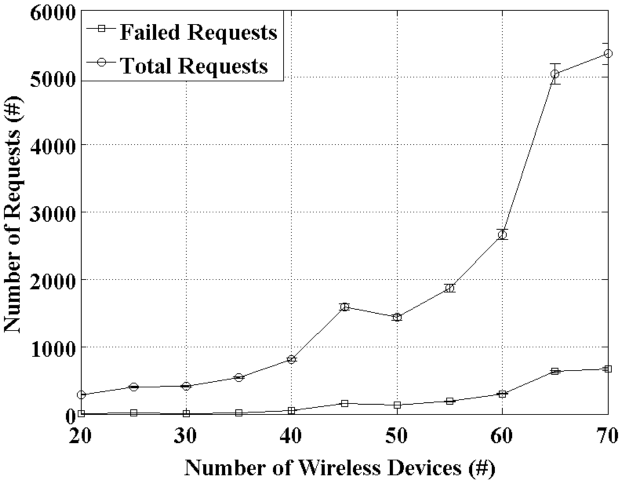


Figure 3. Number of requests with the number of mobile devices participating in the evaluated area.

TABLE II. CLOUD RACK TERMINALS CHARACTERISTICS.

Device #	CPU (GHz)	RAM (GB)	Core No.	Hard Disk (GB)	Cache (MB)	Core Speed (GHz)	Upload Speed (Mbits/sec)
1	2.1	8	Intel Duo	600	2	5	0.6
2	2.3	16	Quad 6600	500	2	5	2.6
3	2.1	4	i5	400	2	3	2.6
4	4.0	16	i5	1000	2	5	2.6
5	2.1	32	i7	600	2	3	4.6
6	2.3	16	i5	500	2	5	2.6
7	2.1	4	Quad 6600	400	2	3	1.5
8	4.0	16	i5	1000	2	5	2.6

On the other hand, the packet drop ratio of the proposed scheme for different mobility variations and without mobility over time is shown in Fig. 6. It is important to emphasize to the fact that the proposed scheme scales well in the presence of FBM and even better when the FBM with distance broadcasting is applied. Fig. 7 presents the average lifetime for both active and idle time with the number of the mobile devices, participating in the evaluated area. The overall energy consumption for each mobile device for three different schemes in the evaluated area (for the most energy draining resources) is shown in Fig. 8. The proposed scheme shows that it outperforms the scheme proposed in [1], as well as the scheme in [7] for the Wi-Fi/WLAN connectivity configuration.

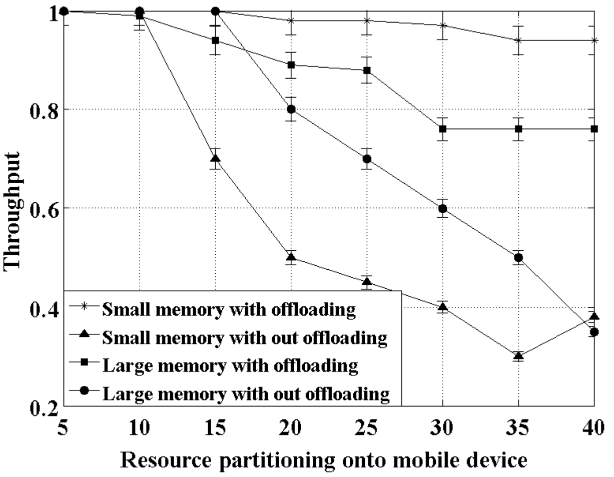


Figure 4. Throughput response with the mean number of executable resources that are partitioned per mobile device.

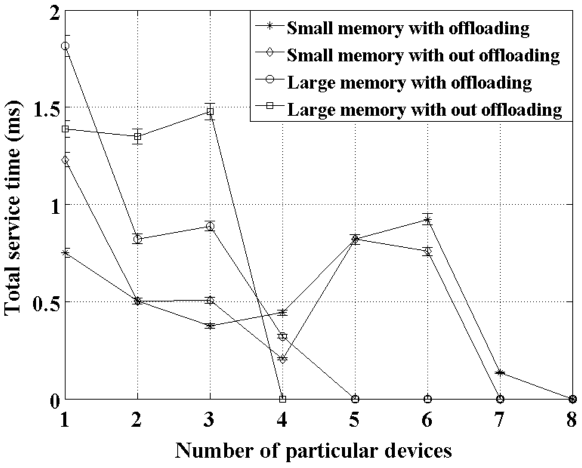


Figure 5. Number of requests with the number of mobile devices participating in the evaluated area.

When resources are offloaded, a critical parameter is the execution time, while nodes are moving from one location to another. In Fig. 9, the execution time during simulation for mobile nodes with different mobility patterns is evaluated, for GSM/GPRS, Wi-Fi/WLAN and for communication within a certain Wi-Fi/WLAN to another Wi-Fi/WLAN remotely hosted. The latter scenario -from a Wi-Fi/WLAN to another Wi-Fi/WLAN- shows to exhibit significant reduction, in terms of the execution time duration, whereas it hosts the minimum execution time through the FBM with distance broadcast mobility pattern. This is due to the access/propagation technology that is used, playing a major role and enabling faster connection, as well as higher transfer rates.

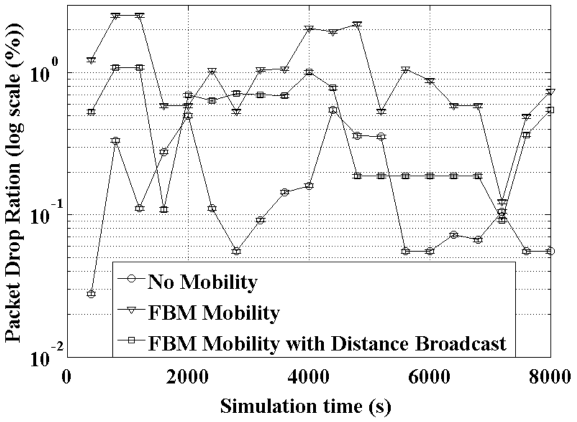


Figure 6. Packet drop ratio of the proposed scheme for different mobility variations and no-mobility model over time.

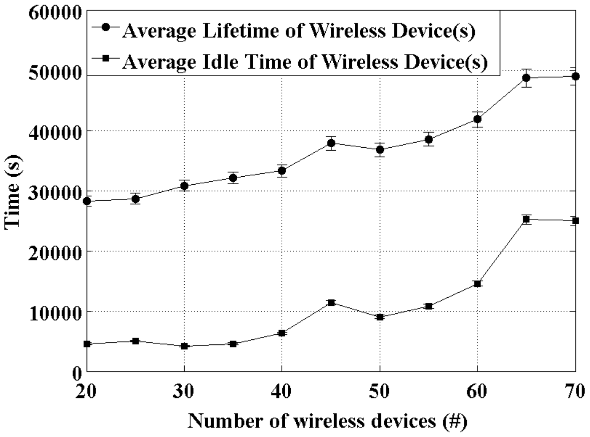


Figure 7. Average lifetime for both active and idle time with the number of mobile devices participating in the evaluated area.

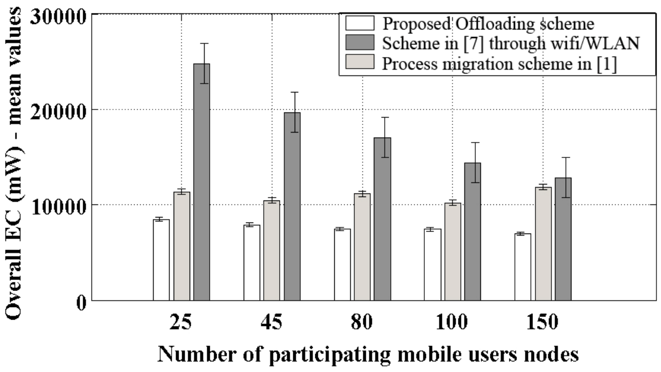


Figure 8. Overall energy consumption for each mobile device for three different schemes in the evaluated area (evaluated for the most energy draining resources).

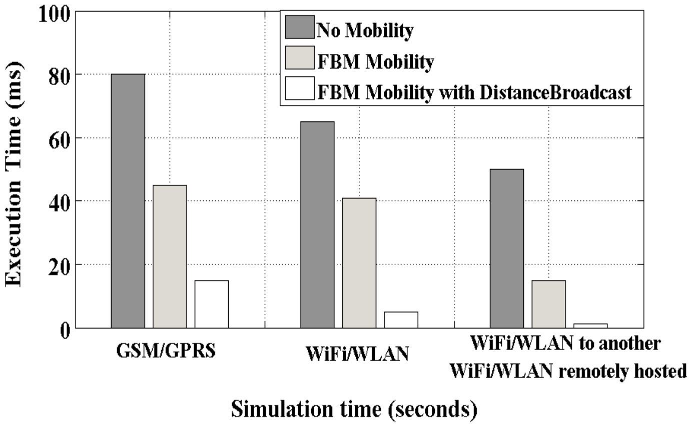


Figure 9. Execution time during simulation for nodes with different mobility patterns for three different schemes of communication.

V. CONCLUSIONS

This paper proposes a novel task outsourcing mechanism comprising of an executable resource offloading scheme. In the proposed scheme partitionable resources can be offloaded, in order to be executed according to their limited service, the transfer time, as well as the allowed execution (round-trip) duration, during the communication with the cloud terminal. The proposed scheme targets the minimization of the energy consumption and the maximization of the lifetime of each mobile device based on the available resources. The proposed offloading scheme is thoroughly evaluated through simulation experiments to validate the efficiency of the offloading policy, in contrast to the energy consumption of the mobile devices, as well as for the reliability degree offered. Future directions in our on-going research encompass the improvement of an opportunistically formed mobile cloud, which will allow delay-sensitive resources to be offloaded using the mobile peer-to-peer (MP2P) technology.

ACKNOWLEDGMENT

The work presented in this paper is co-funded by the European Union, Eurostars Programme, under the project 8111, DELTA "Network-Aware Delivery Clouds for User Centric Media Events".

REFERENCES

[1] P. Mousicou, C. X. Mavromoustakis, A. Bourdena, G. Mastorakis, and E. Pallis, "Performance evaluation of Dynamic Cloud Resource Migration based on Temporal and Capacity-aware policy for Efficient Resource sharing, accepted to MSWiM, The 16th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, November 3-8 2013, Barcelona, Spain, pp. 59-66.

[2] S. Abolfazli, Z. Sanaei, E. Ahmed, and A. Gani, R. Buyya, "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies", and Open Challenges. IEEE Communications Surveys and Tutorials 16(1), 2014, pp. 337-368.

[3] C. Dimitriou, C.X. Mavromoustakis, G. Mastorakis, and E. Pallis, "On the performance response of delay-bounded energy-aware bandwidth allocation scheme in wireless networks", in Proceedings of the IEEE/ICC 2013 International Workshop on Immersive & Interactive Multimedia Communications over the Future Internet, organized in conjunction with IEEE International Communications Conference (ICC 2013), 9-13 June 2013, Budapest, Hungary, pp.641-646.

[4] M. A. Salehi, B. Javadi, and R. Buyya, "Resource provisioning based on preempting virtual machines in distributed systems". Concurrency and Computation: Practice and Experience 26(2), 2014, pp. 412-433.

[5] C. X. Mavromoustakis, C. Dimitriou, G. Mastorakis, E. Pallis, "Real-Time Performance Evaluation of F-BTD scheme for optimized QoS Energy Conservation in Wireless Devices", in Proceedings of the GlobeCom 2013 2013, pp. 1156-1161.

[6] D. Warneke and O. Kao, "Exploiting Dynamic Resource Allocation for Efficient parallel data processing in the cloud". IEEE Transactions on Parallel and Distributed Systems, VOL. 22, No. 6, June 2011, pp.985-997.

[7] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. "Clonecloud: Elastic execution between mobile device and cloud". Proceedings of the sixth conference on Computer systems of EuroSys, pp.301-314, 2011.

[8] B. G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution". In HotOS, 2009.

[9] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To Offload or Not to Offload? The Bandwidth and Energy Costs of Mobile Cloud Computing", Proc. of INFOCOM 2013, April 2013, Turin, Italy.

[10] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," Proc. of the ACM International Conference on Mobile Systems, Applications, and Services, pp.49-62, San Francisco, CA, USA, 15-18 June 2010, pp. 647-652.

[11] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing". Pervasive Computing, 8(4), 2009, pp. 14-23.

[12] V. K.Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability". Proceedings of the 1st ACM Symposium on Cloud Computing, June, pp. 193-204, 2010.

[13] C. X. Mavromoustakis, C. D. Dimitriou and G. Mastorakis, "On the Real-Time Evaluation of Two-Level BTD Scheme for Energy Conservation in the Presence of Delay Sensitive Transmissions and Intermittent Connectivity in Wireless Devices", International Journal On Advances in Networks and Services, Vol. 6 number 3&4 2013, pp. 148-162.

[14] K. Papanikolaou and C. X. Mavromoustakis, "Resource and scheduling management in Cloud Computing Application Paradigm", appears in book "Cloud Computing: Methods and Practical Approaches", Eds. Prof. Zaigham Mahmood, published in Methods and Practical Approaches Series: Computer Communications and Networks by Springer International Publishing, May 13, 2013, ISBN 978-1-4471-5106-7, pp. 107-132.

[15] A. Bourdena, C. X. Mavromoustakis, G. Kormantzas, E. Pallis, G. Mastorakis, M. B. Yassein, "A Resource Intensive Traffic-Aware Scheme using Energy-efficient Routing in Cognitive Radio Networks", Future Generation Computer Systems Journal, Elsevier; to appear during the year 2014.

Approach on Fraud Detection in Voice over IP Networks using Call Destination Profiling Based on an Analysis of Recent Attacks on Fritz!Box Units

Anton Wiens, Torsten Wiens and Michael Massoth
Department of Computer Science
Hochschule Darmstadt – University of Applied Science
Darmstadt, Germany
{anton.wiens | torsten.wiens | michael.massoth}@h-da.de

***Abstract*—Recently, massive attacks on Fritz!Box hardware units have been disclosed, caused by a security vulnerability. The Fritz!Box by AVM is an multifunctional routing device, offering Voice over IP-(VoIP) and internet connectivity to private users, which is in wide use in Germany. By first taking over the units and in a second step using the units to conduct toll fraud attacks on VoIP providers and their customers, significant financial damage has been caused. In this work, these attacks are analyzed and attack patterns as well as their characteristic traits are described. Based on these results, a novel method for toll fraud detection is devised and evaluated. The method is capable of detecting this kind of attack, as well as similar attack patterns. Results of a prototype implementation show successful detection of these attacks, enabling to prevent them in the future. This work is based on real-life traffic data from a cooperating telecommunication service provider.**

Keywords—Fraud Detection; Voice Over IP Networks; Fritz!Box; User Profiling; Statistical detection methods.

I. INTRODUCTION

Today's voice communication by Voice over IP (VoIP) mostly uses the internet for data transport. There are the drawbacks that the internet can basically be accessed by anyone, and that it links anyone to anyone. For example, it is possible for third parties with criminal intent to access private branch exchange (PBX) systems connected to the internet.

Fraudsters may have multiple options to abuse these systems. Systems that are insufficiently secured may be tapped. Access data that has been saved in these systems could be used to abuse, compromise or even take over the whole PBX. If the PBX system has been taken over, a fraudster will be able to conduct telephone calls to premium rate service numbers or comparable call destinations, generating profit. The resulting cost, on the other hand, will often be charged to the victim or its telecommunication service provider, because of a general rule in telecommunication service providing, called "Calling Party Pays".

The Communications Fraud Control Association (CFCA) reports losses of about 46 billion USD caused by telecommunication fraud in 2013, an increase by 15% compared to 2011 [1]. Not only financial damage is a problem caused by fraud attacks. Small providers may also suffer from reputation losses, causing customers to change the provider because of decreased trust and fear of repeated fraud attempts in the future.

To detect and counter these attacks, respectively fraud attempts, fraud detection systems are used. Often, these systems apply methods based on the generation of statistical profiles for each user. User profiles are generated that describe their behavior. These profiles will then be used as input for machine learning techniques, allowing for the detection of fraud [2] [3][4] [5].

The German company "Deutsche Telekom" reported a huge success in the prevention of fraud cases with potential damages of about 200 million Euro, using an automated fraud detection system [6]. The research project "Trusted Telephony" at the University of Applied Sciences Darmstadt, from which the work at hand originates, pursues the goal to increase security in VoIP telephony, cooperating with the German telecom service provider toplink GmbH. A key objective of the project is the development of a fraud detection system.

Recently, fraud cases were caused by security exploits in Fritz!Box hardware (from the company AVM GmbH), which is often used in Germany [7][8]. The Fritz!Box is an integrated, multifunctional routing device, offering internet connectivity, VoIP capabilities and other services in local area networks. This unit is very popular in Germany. Because of the large amount of units in use, there is an increased risk in case of security vulnerabilities, especially for private users.

On the other hand, an exploitation of the recently disclosed security vulnerability of this unit is only one possibility to start such attacks. The security vulnerability has been patched by the manufacturer in the mean time, but in the future, comparable vulnerabilities in similar hardware could turn up. Therefore, it is important to be able to detect these situations and devise measures to counter them.

In the work at hand, an analysis of the recent attacks on Fritz!Box units is presented. Characteristic traits of these attacks are described, classified and analyzed. Additionally, it is discussed if the usual methods for the detection of fraud cases are also applicable in these cases. Resulting from this analysis, a new fraud detection method is devised. The basic idea is not to apply a variant of user profiling techniques, as usual, but to use statistical profiles of call destination numbers. This way, it is possible to detect certain attacks that would go undetected if user profiling techniques were applied. The general problem with this kind of attack pattern is a distribution of single attacks over multiple users, whose router units have been compromised by a primary attack. Therefore, this pattern cannot be detected as a fraud attack from the perspective of a single user.

The new method uses two profiles, as described in our preliminary work for a different case [9]. These profiles are used to describe the behavior of destination call numbers in defined time spans in the past and present. Changes in behavior are statistically evaluated. Fraud attempts are detected by the investigation of major changes in this data.

A. *Call detail records*

The data being analyzed in this work comprises fraud attacks that have been enabled by the recently discussed security vulnerability of the Fritz!Box units. The data, consisting of Call Detail Records (CDR) has been supplied by toplink GmbH.

A CDR is a text file, containing all parameters of single telephone calls. Each CDR is written by the primary VoIP routing system TELES.iSwitch at toplink as calls are set up [10]. CDRs contains information on caller, callee, call duration, starting time as well as technical network parameters.

B. *Structure of the Paper*

After the introduction, an overview of related work is given in Section II, its relevance is described as well. In Section III, the concept of behavior profiling is introduced, on which the method presented in this paper is based. In Section IV, Call Detail Records are introduced. Section V contains an analysis of attacks on telecommunication systems that are enabled by the known security vulnerability of Fritz!Box hardware. In Section VI, a method to identify and counter such attacks is presented. First evaluation results of this method, based on real-life CDR traffic data, are presented in Section VII. Section VIII contains a conclusion to this paper, and Section IX presents possible future work.

II. RELATED WORK

This work is partially based on findings from preliminary work of the authors [9], as well as additional ideas that arose during the recent announcement of the attacks on Fritz!Box hardware [8].

In [9], a method for toll fraud detection using statistical user profiling has been described, which can especially be applied when no significant amount of training data is available. Additionally, the method can be run in a mostly autonomous way, requiring only a minimum amount of external administration. The method applies two user profiles, one for a past period of time and one for a present period of time, each containing statistical features. The profiles are used to identify suspicious deviations of the users' behavior, by which toll fraud attempts are detected. In this work, the attacks on Fritz!Box hardware and the possibility to detect these using the presented method had already been mentioned.

In the work at hand, the method from the preliminary work is adapted more closely to this attack pattern. The new method again uses two profiles of statistical features for each user, but differing in contents and their actual use for the detection of attacks.

Furthermore, other related work also describes different methods of user profiling for the detection and prevention of toll fraud in VoIP telecommunication [2] [3] [5] [11] [12] [13]. In contrast to this work, the work at hand does not apply simple user profiles, but a new kind of profile specified as Call Destination Profile. These profiles are used to characterize the behavior of a destination telephone number instead of a user's behavior. It is intended to detect special kinds of attacks this way.

These attacks cannot be detected with user profiling techniques alone and hence would go undetected if the method from [9] was applied. Section V contains a more in-depth description of the idea.

III. BEHAVIOR PROFILING

The term „behavior profiling" describes a technique for differential analysis where the behavior of a given object is represented by a statistical profile. In the profile, data from the object is accumulated, which is then used to generate statistics that describe the object's behavior, which are called features. Often, behavior profiles are applied in the form of user profiles [2] [3] [5] [11] [12] [13]. In most cases, a differential analysis is preferred over an absolute analysis. This is because the absolute analysis is a subset of the differential analysis [9].

For example, three variants of user profiling methods are presented in [4]. In this work, the parameters *duration per call*, *number of calls per customer* and *costs per call* are arranged in different ways into the group's *national calls*, *international calls* and *mobile calls.* These are used to generate statistics for the profiles.

User profiles are utilized to describe the behavior of users in the present and in the past, enabling a comparison of behavioral patterns. By this comparison, it is possible to detect suspicious fluctuations. These are analyzed in the next step in order to generate a decision on fraudulent or non-fraudulent behavior.

In the work at hand, behavior profiling is applied for a novel profiling approach, differing from classical user profiling in the way that no profiles of the user's behavior are generated, but profiles of destination call numbers instead.

IV. ANALYSIS OF ATTACKS ON FRITZ!BOXES

The recent attacks at (and by) Fritz!Boxes can be divided in two categories. The first category comprises the hostile takeover of a Fritz!Box by exploiting a security vulnerability in its firmware. The second category comprises possible results of such a takeover, especially secondary attacks that are enabled by then remotely controllable units. Both categories are described in more detail in the following subsections. It is important to note that the initially possible attacks on these units cannot be conducted anymore, since the firmware has been updated by the manufacturer in the mean time [8]. The focus of the work at hand is at the possibility of fraud attacks on telecommunication systems by utilizing taken over secondary hardware, which is not unlikely to happen again in the future, and detecting it.

A. *Primary hostile take-over of a Fritz!Box*

The basic idea to perform a hostile take-over of a Fritz!Box was as follows: An attacker would set up a web site, which is to be visited by potential victims. The attacker would then be able to exploit the known security vulnerability of the Fritz!Box in order to extract the master password. Using this password, the attacker would be able to access the command shell. Once this is done, the attacker could then deploy system commands, e.g., to make the unit call premium-rate service numbers at the cost of the unit's owner [8].

B. *Secondary attacks after the take-over*

Attack attempts on other systems that had been conducted using taken-over Fritz!Boxes seem to be very similar in their basic approach. For an in-depth analysis, anonymized data on such attack attempts has been provided by toplink GmbH. The data being used is in accordance to the Federal German Data Protection Act (Bundesdatenschutzgesetz) [14]. All results from this analysis are based on this data and may not represent attack patterns that appeared at other telecommunication providers.

1) *From a single user's view*

From the perspective of a single user, an attack attempt may look as follows: An attacker tries to set up a call to a premium-rate service number or a comparably expensive call destination, possibly also in another country. This is done multiple times during a short time span. As soon as the attacker has successfully set up a call to a given number, he will try to call this number again, as often as possible, and also in a short period of time. If the call attempts fail (e.g., because the number is not available), the attacker will try another number,

The difficulty to detect such attack attempts lies in the low frequency and the low duration of these calls seen from a single user's point of view. Attackers will avoid a detection using these two parameters by applying an approach described in the next section.

2) *Exploiting multiple users*

By exploiting the security vulnerability at multiple victims' Fritz!Box units, attackers are able to hide their attack attempts neatly. The attack attempts are distributed across multiple taken-over units. So, it becomes possible to mask obvious evidence of attack attempts, such as frequency and duration of calls. This will be illustrated by the following examples:

1. Attacker A conducts a hostile take-over of victim C and causes C's unit to start 30 calls to destination number B. The duration of each call is 20 seconds.
2. Attacker A conducts a hostile take-over of victim C and causes C's unit to start 5 calls to destination number B. The duration of each call is 5 minutes.
3. Attacker A conducts hostile take-overs of 30 victims and causes each victim's unit to conduct one call to destination number B. The duration of each call is 20 seconds.

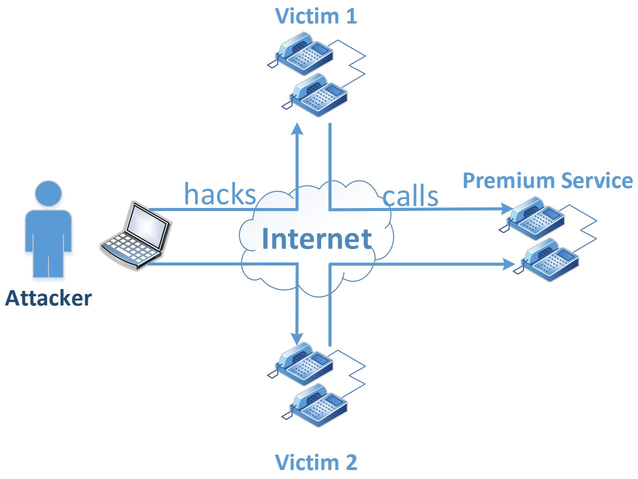


Figure 1. Depiction of example three with just two victims of an attacker calling a premium service number

In the first example, the attack at victim C can be detected by the frequency of the calls. In the second example, the attack can be detected by the extraordinarily long duration of the calls. In the third example, the features used before cannot be used again. Existing methods often apply user profiling to detect suspicious behavior and potential attack- or fraud cases. This way, distributed attacks as described in example three, cannot be detected. Therefore, it is necessary to apply a different method for detection. Figure 1 shows a depiction of example three with just two victims of an attacker calling a premium service number.

Existing methods often apply user profiling to detect suspicious behavior and potential attack- or fraud cases. This way, distributed attacks as described in example 3, cannot be detected. Therefore, it is necessary to apply a different method for detection.

C. *Characteristic traits for detection*

From the results of the preceding section, the following characteristic traits for detection can be deduced:

- **Duration of call for a certain user**: The call duration is significantly higher in comparison to the known behavior of that user.
- **Number of calls for a certain user**: The number of calls in a given time span is significantly higher in comparison with the known behavior of that user
- **Number of calls for a certain destination number:** The number of calls that has been conducted to a given (premium rate service-) destination number in a given time span is suspicious

The first two of these characteristic traits can be detected by applying user profiling if the perspective of a single user is applied. To be able to detect attack attempts using the number of calls, a new method has to be devised. This will be described in Section VI.

V. Basic Concept of Detection Method

Because of the large amount of call attempts that are distributed to many users, it is necessary to devise a profiling method that does not differentiate single users, but destination numbers. This method is named *Call Destination Profiling* and will be described in this section.

Call Destination Profiling differentiates single Call Detail Records (see Section IV) by destination number. Compared to user profiling, a destination number is been looked upon as a “user”. For each destination number, two profiles are generated and analyzed to detect attack- or fraud attempts. The *Past Behavior Profile (PBP)* is used to describe the behavior in the past. The *Current Behavior Profile (CBP)* is used to describe the behavior in the present.

A. Profile

In this method, a profile describes the behavior of a destination number and not the behavior of a user. Because it is possible for different users with different behavioral patterns to conduct calls to a given destination number, it is not possible to use the same behavior-describing statistics (features) as in user profiling. In user profiling, it is often the case to collect statistical data on the duration and the frequency of calls [2] [4] [5] [11] [15]. Since different callers may conduct calls of different length, it is less reasonable to collect statistical data on the duration of calls.

As mentioned in Section V, the number of calls to certain destination numbers represents an important feature for profiles used to detect the described attacks. For this reason, the following features are used in the PBP:

- Arithmetic mean of the number of calls per hour *(MeanCalls)*
- Standard deviation of the number of calls per hour *(StdCalls)*

In addition to those features, the time span t_{PBP} that the PBP will comprise has to be determined. If t_{PBP} is too long, it will take longer to initialize it with data. On the other hand, it will offer more robust statistics.

If t_{PBP} is too short, the statistics describing the past behavior may be not robust enough, possibly introducing inaccuracies into the detection process.

Based on findings in our preliminary work [9], the following rules apply for the construction of profiles: If t_{PBP} comprises a time span of less than one week, then “gaps” in the statistics will result. These gaps will cause large deviations of the measurements in the accumulated statistics. Furthermore, if the profiles are too short, e.g., one single day, large deviations will also occur. This is because the number of measurements is too low. For this reason, t_{PBP} will be set to one week, as it has also been done in our preliminary work.

In contrast to the preliminary work, a different profile time span t_{CBP} will be applied for the CBP. This is justified by the use of a different comparison function, which is described in more detail in the following Section. For the CBP, especially the number of unique callers *(NumCallees)* and the number of calls *(NumCalls)* is of great relevance. The length of the CBP determines the effects of individual fraud cases on the statistics of the profile. If *MeanCalls* and *StdCalls* relate to calls per hour, a length of one hour is determined by this. Longer or shorter profiles are more suitable for fraud attempts that are spread farther or closer on the time scale. For the time being, a profile time span t_{CBP} of one hour was applied.

Figure 2 displays a simplified diagram of both profiles in relation to time, as well as the features used.

B. Comparison of profiles and fraud detection

As already mentioned, the comparison function for the profiles differs from the function used in our preliminary work.

$$CallLimit = MeanCalls + StdCalls \cdot G_R + A_R \quad (1)$$

If CBP comprises a shorter time span as the PBP, the comparison can obviously not be conducted in the same way as before. The comparison will now be done in the following way:

A threshold *CallLimit* for each destination number is calculated from the features *MeanCalls* and *StdCalls* of the PBP using (1). A weighting parameter G_R has been introduced for the feature *StdCalls*, to allow for a finer adjustment of the relative component $MeanCalls + StdCalls$. Additionally, an absolute component $A_R$has been added to enable the analysis of infrequently called destination numbers. This component is used to compensate for errors as well, as long as the profile is still empty. Furthermore, the absolute component A_R and the weighting parameter G_R have to be selected depending on the geographic destination region *R* of the destination number. Region *R* is distinguished into national calls, mobile calls and international calls. This is to allow for a different treatment of national, mobile and international calls, each causing different costs, and differing in regard to potential damages from the perspective of telecommunication service providers.

The threshold *CallLimit* is finally compared to the feature *NumCalls* within the CBP to decide upon fraudulent or non-fraudulent behavior using (2).

$$Fraud = \begin{cases} true, & NumCalls \geq CallLimit \\ false, & NumCalls < CallLimit \end{cases} \quad (2)$$

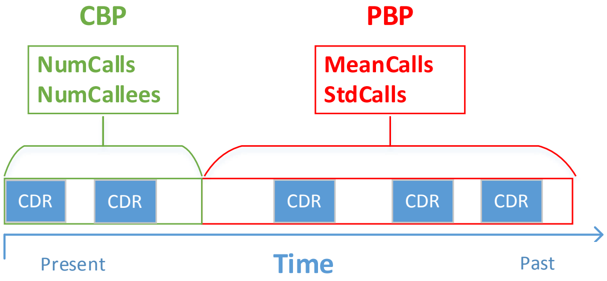


Figure 2. Depiction of the current behavior profile and the past behavior profile in relation to time

If the threshold is exceeded, the related call will be marked as fraudulent. A detection using the number of callers *(NumCallees)* can be conducted as an option. For detection, the number of calls is the most important parameter, which is very meaningful for administrative staff to support their final decision in the process.

VI. PROTOTYPE

In this section, results from a prototype implementation of the devised method are described and analyzed empirically. In Subsection A, the data set in use is specified. Subsection B describes the experimental setup. The final results are presented in Subsection C.

A. *Used Data Set*

To evaluate the prototype implementation, real life traffic data (CDRs) provided by toplink GmbH has been used. The data comprises calls from a time span of two weeks containing about 3.5 million calls. Only the portion of the data with outgoing calls was used, because incoming calls are not relevant to the analysis. The outgoing calls amount to about 470,000. Table I shows the distribution of the calls for the regions national, mobile and international and are split into connected and unconnected calls.

TABLE I. NUMBER OF CDRS FOR EACH REGION

REGION	AMOUNT
CONNECTED	325,947
NATIONAL	274,205
MOBILE	42,669
INTERNATIONAL	9,073
UNCONNECTED	153,330
NATIONAL	112,476
MOBILE	24,570
INTERNATIONAL	16,284
TOTAL	479,277

In the first week, no attack attempts (fraud) were contained. This part of the data was applied to initialize the behavior profiles, building the features. In the second week, normal call traffic is contained as well as about 20,140 fraudulent calls following the typical Fritz!Box attack pattern. The second week has been used to test the detection abilities.

B. *Experimental Setup*

First of all, the relevant thresholds had to be determined, because this is a necessity for high-quality detection results. To accomplish this, a single run of the method, without the fraud detection, is conducted with the first week of the data and every feature value at the time of each call is recorded. The thresholds are estimated by analyzing the resulting values of the CBP for fraud and non-fraud cases and for each region (national, mobile, international). The 99%-quantiles of the number of calls from the CBP, for connected and unconnected calls as well as national, international and mobile calls each, have been recorded and used as the absolute threshold A_R for each region. The parameter G_R, representing the relative threshold, has been set to $G_R = 1$, for testing purposes.

Finally, a test run with the activated fraud detection and the previously measured thresholds is done and the detection quality is evaluated by comparing the detected cases to the known cases of fraudulent behavior.

The approach can be described with the following steps:

1. The detection method is deactivated at first
2. The profiles are initialized using the first week data set
3. Thresholds are calculated from CBP values as described before
4. The detection method is now activated
5. The second week data set is now used as input
6. The results from the detection method are compared to the known cases of fraudulent behavior

C. *Results*

Thresholds have been determined for successfully connected as well as unconnected call attempts, each for national, international and mobile calls. Also, the profile values have been calculated and recorded.

Unfortunately, the thresholds determined herein cannot be published for security reasons. This would especially allow fraud attackers to refine their attack patterns. On the other hand, the thresholds in this case represent the actual test data and wouldn't be representative for the situation at other telecommunication service providers. For this reason, only the results for the detection method are described.

The arithmetic mean and the standard deviation both represent valid values to generate relative thresholds, as mentioned in Section VI-B. An adjustment with the parameter G_R is only necessary in individual cases.

Under these testing conditions, the detection method achieved a false positive rate of 0.7% or 3,355 false positives (as show in Table II). Of the known attacks in the data, the detection method was able to identify all attacks, resulting in 100% detection rate or true positive rate. However, there is the possibility that not all attacks are detected because some may be unknown. An estimation of a true positive rate of about 95% would be more appropriate.

TABLE II. DETECTION RESULTS

	AMOUNT	RATE
FALSE POSITIVE	3,355	0.7%
TRUE POSITIVE	20,140	100%

Compared to the results achieved in comparable related work (see Table III), which utilizes unsupervised user profiling, with a FPR of 4% and a TPR of 75% [3] and our previous work with a FPR of 1.22% and a TPR of about 90% [9], these measurements are as good or even better.

ISBN: 978-1-61208-357-5

TABLE III. COMPARISON OF FPR AND TPR

	TPR	FPR
THIS WORK	95%	0.7%
PREVIOUS WORK [9]	90%	1.22%
RELATED WORK [3]	75%	4%

On the other hand, no direct comparison is possible, because the detection method itself is partially different, applying a modified approach of user profiling. Additionally, the number of callees *(NumCallees)* has been found a viable criterion for administrative staff to make decisions on fraudulent or non-fraudulent behavior.

VII. CONCLUSION

The presented method successfully detects distributed fraud attacks that are conducted using multiple Fritz!Box units. The test results show that the method may also be applied to similar attack patterns in the future. It has to be stressed that the focus of this work has been on devising a universally applicable method rather than a specialized one, because the security vulnerability in the Fritz!Box has since been patched, but it is possible to use different hardware units for similar attack patterns in the future.

The method offers a low false positive rate. An experimental evaluation showed that all known cases of fraud attacks in the test data were detected.

VIII. FUTURE WORK

In future work, a further evaluation of call destination profiles will be done.

For example, neural networks could be applied as well, as in [15], delivering comparable results. Neural networks are not as transparent as basic statistical approaches, such as used in the presented method. Other techniques may also be applicable, for example support vector machines (SVM). On the other hand, significantly more data than in the presented case would be necessary for training. Since the training data should not contain fraud cases, the effort for generation from real life traffic data would be high. The presented method is also to be integrated into the fraud detection framework developed within this research project.

ACKNOWLEDGMENT

We would like to thank the state of Hessen, Germany for supporting this work by providing the necessary funds by the development program LOEWE. Also we would like to thank toplink GmbH from Darmstadt, Germany for cooperating with us in our project and providing essential support, data and knowledge for research and development of practical fraud detection methods. We would also like to thank Sandra Kübler for her valuable comments and general input on the subject.

REFERENCES

[1] Communications Fraud Control Association, "Global Fraud Loss Survey," October 2013. [Online]. Available: http://www.cfca.org/pdf/survey/CFCA2013GlobalFraudLossSurvey-pressrelease.pdf. [retrieved: 6, 2014].

[2] M. Taniguchi, M. Haft, J. Hollmen, and V. Tresp, "Fraud detection in communication networks using neural and probabilistic methods," in Proceedings of the 1998 IEEE International Conference on : Acoustics, Speech and Signal Processing, vol. 2, 1998, pp. 1241-1244.

[3] P. Burge and J. Shawe-Taylor, "Detecting Cellular Fraud Using Adaptive Prototypes," in Proceedings AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management, AAAI Press, 1997, pp. 9-13.

[4] C. S. Hilas and P. A. Mastorocostas, "An application of supervised and unsupervised learning approaches to telecommunications fraud detection," Knowledge-Based Systems, vol. 21, no. 7, pp. 721-726, 2008.

[5] H. Grosser, P. Britos, and R. García-Martínez, "Detecting fraud in mobile telephony using neural networks," in Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence, Bari, Italy, Springer-Verlag, 2005, pp. 613-615.

[6] heise online, "Bericht: Deutsche Telekom wertet Verbindungsdaten sämtlicher Telefonate aus | heise online," [Online]. Available: http://www.heise.de/newsticker/meldung/Bericht-Deutsche-Telekom-wertet-Verbindungsdaten-saemtlicher-Telefonate-aus-1933436.html. [retrieved 6, 2014].

[7] AVM GmbH, 06 02 2014. [Online]. Available: https://www.avm.de/de/News/artikel/2014/sicherheitshinweis_telefonmissbrauch.html. [retrieved 6, 2014].

[8] R. Eikenberg, "Hack gegen AVM-Router: Fritzbox-Lücke offengelegt, Millionen Router in Gefahr," 07 03 2014. [Online]. Available: http://www.heise.de/security/meldung/Hack-gegen-AVM-Router-Fritzbox-Luecke-offengelegt-Millionen-Router-in-Gefahr-2136784.html. [retrieved 6, 2014].

[9] A. Wiens, T. Wiens, and M. Massoth, "A new Unsupervised User Profiling Approach for Detecting Toll Fraud in VoIP Networks," in The Tenth Advanced International Conference on Telecommunications, to be published 2014.

[10] TELES AG, [Online]. Available: http://www.teles.com/en/teles.html. [retrieved 6, 2014].

[11] Y. Moreau, H. Verrelst, and J. Vandewalle, "Detection of Mobile Phone Fraud Using Supervised Neural Networks: A First Prototype," in Proceedings of the 7th International Conference on Artificial Neural Networks, Springer-Verlag, 1997, pp. 1065-1070.

[12] T. Kapourniotis, T. Dagiuklas, G. Polyzos, and P. Alefragkis, "Scam and fraud detection in VoIP Networks: Analysis and countermeasures using user profiling," in FITCE Congress (FITCE), 2011 50th, 2011, pp. 1-5.

[13] T. Fawcett and F. Provost, "Adaptive Fraud Detection," Data Mining and Knowledge Discovery, vol. 1, no. 3, pp. 291-316, 1997.

[14] "Der Bundesbeauftragte für den Datenschutz und die Informationsfreiheit: Bundesdatenschutzgesetz (BDSG)," [Online]. Available: http://www.bfdi.bund.de/cae/servlet/contentblob/409518/publicationFile/25234/BDSG.pdf. [retrieved 6, 2014].

[15] P. Burge, J. Shawe-Taylor, C. Cooke, Y. Moreau, B. Preneel, and C. Stoermann, "Fraud detection and management in mobile telecommunications networks," in European Conference on : Security and Detection. ECOS 97., 1997, pp. 91-96.