ICSNC 2017

The Twelfth International Conference on Systems and Networks Communications

ISBN: 978-1-61208-591-3

October 8 - 12, 2017

Athens, Greece

ICSNC 2017 Editors

Eugen Borcoci, University Politehnica of Bucarest, Romania

Renwei (Richard) Li, Future Networks, Huawei, USA

Juan Jose Vegas Olmos, Mellanox Technologies, Denmark

ICSNC 2017

Forward

The Twelfth International Conference on Systems and Networks Communications (ICSNC 2017), held on October 8 - 12, 2017- Athens, Greece, continued a series of events covering a broad spectrum of systems and networks related topics.

As a multi-track event, ICSNC 2017 served as a forum for researchers from the academia and the industry, professionals, standard developers, policy makers and practitioners to exchange ideas. The conference covered fundamentals on wireless, high-speed, mobile and Ad hoc networks, security, policy based systems and education systems. Topics targeted design, implementation, testing, use cases, tools, and lessons learnt for such networks and systems

The conference had the following tracks:

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

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

We take here the opportunity to warmly thank all the members of the ICSNC 2017 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the ICSNC 2017. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ICSNC 2017 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the ICSNC 2017 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in networking and systems communications research. We also hope Athens provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful historic city.

ICSNC Steering Committee

Eugen Borcoci, University "Politehnica"of Bucharest (UPB), Romania
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Leon Reznik, Rochester Institute of Technology, USA
Zoubir Mammeri, IRIT - Paul Sabatier University, Toulouse, France
Maiga Chang, Athabasca University, Canada
David Navarro, Lyon Institute of Nanotechnology, France
Christos Bouras, University of Patras / Computer Technology Institute & Press 'Diophantus', Greece

ICSNC Industry/Research Advisory Committee

Yasushi Kambayashi, Nippon Institute of Technology, Japan
Christopher Nguyen, Intel Corp., USA

ICSNC 2017

Committee

ICSNC Steering Committee
Eugen Borcoci, University "Politehnica"of Bucharest (UPB), Romania
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Leon Reznik, Rochester Institute of Technology, USA
Zoubir Mammeri, IRIT - Paul Sabatier University, Toulouse, France
Maiga Chang, Athabasca University, Canada
David Navarro, Lyon Institute of Nanotechnology, France
Christos Bouras, University of Patras / Computer Technology Institute & Press 'Diophantus', Greece

ICSNC Industry/Research Advisory Committee
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Christopher Nguyen, Intel Corp., USA

ICSNC 2017 Technical Program Committee

Habtamu Abie, Norwegian Computing Center - Oslo, Norway
Talal Alharbi, University of Queensland, Australia
G. G. Md. Nawaz Ali, Nanyang Technological University (NTU), Singapore
Samr Samir Ali, Abu Dhabi University, UAE
Muhammad Sohaib Ayub, Lahore University of Management Sciences (LUMS), Pakistan
K. Hari Babu, Birla Institute of Technology & Science (BITS Pilani), India
Ilija Basicevic, University of Novi Sad, Serbia
Robert Bestak, Czech Technical University in Prague, Czech Republic
Eugen Borcoci, University "Politehnica"of Bucharest (UPB), Romania
Badre Bossoufi, Higher School of Technology, EST-Oujda, Morocco ,
Christos Bouras, University of Patras / Computer Technology Institute & Press <Diophantus>, Greece
Martin Brandl, Danube University Krems, Austria
Fernando Brito e Abreu, Instituto Universitário de Lisboa (ISCTE-IUL), Portugal
Dumitru Dan Burdescu, University of Craiova, Romania
Vicente Casares Giner, Universitat Politècnica de València, Spain
Maiga Chang, Athabasca University, Canada
Hao Che, University of Texas at Arlington, USA
Stefano Chessa, University of Pisa, Italy
Enrique Chirivella-Perez, University of the West of Scotland, UK
Jorge A. Cobb, The University of Texas at Dallas, USA
Bernard Cousin, Irisa | University of Rennes 1, France
Sima Das, MST, USA
Poonam Dharam, Saginaw Valley State University, USA
Gulustan Dogan, Yildiz Technical University, Istanbul, Turkey
Jawad Drissi, Cameron University, USA
Safwan El Assad, University of Nantes, France

Marco Furini, University of Modena and Reggio Emilia, Italy
Pedro Gama, Truewind, Portugal
Katja Gilly, Universidad Miguel Hernández, Spain
Hector Marco Gisbert, University of the West of Scotland, UK
Rich Groves, A10 Networks, USA
Barbara Guidi, University of Pisa, Italy
Youcef Hammal, USTHB University Bab-Ezzouar, Algeria
Xavier Hesselbach-Serra, Universitat Politècnica de Catalunya (UPC), Spain
Chitra Javali, UNSW Sydney, Australia
Muhammad Javed, Cameron University, USA
Fang-zhou Jiang, Data61 | CSIRO & UNSW, Australia
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Sokratis K. Katsikas, Norwegian University of Science & Technology (NTNU), Norway
Jinoh Kim, Texas A&M University-Commerce, USA
Yagmur Kirkagac, Netas Telecommunication Inc., Turkey
Peng-Yong Kong, Khalifa University of Science, Technology & Research (KUSTAR), Abu Dhabi, UAE
Michał Król, University College London, UK
Gyu Myoung Lee, Liverpool John Moores University, UK
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Zoubir Mammeri, IRIT - Paul Sabatier University, Toulouse, France
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Francisco J. Martinez, University of Zaragoza, Spain
Victor Mehmeri, Technical University of Denmark (DTU), Denmark
Farouk Mezghani, Inria Lille - Nord Europe, France
David Navarro, Lyon Institute of Nanotechnology, France
Christopher Nguyen, Intel Corp., USA
António Nogueira, University of Aveiro / Instituto de Telecomunicações, Portugal
Jun Peng, University of Texas - Rio Grande Valley, USA
Zeeshan Pervez, University of the West of Scotland, UK
Kandaraj Piamrat, CReSTIC - University of Reims Champagne-Ardenne, France
Paulo Pinto, Universidade Nova de Lisboa, Portugal
Aneta Poniszewska, Lodz University of Technology, Poland
Victor Ramos, Metropolitan Autonomous University, Mexico
Piotr Remlein, Poznan University of Technology, Poland
Yongmao Ren, Chinese Academy of Sciences, China
Girish Revadigar, UNSW Sydney, Australia
Leon Reznik, Rochester Institute of Technology, USA
Mohsen Rezvani, Shahrood University of Technology, Iran
Laborde Romain, University Paul Sabatier (Toulouse 3), France
Luis Enrique Sánchez Crespo, University of Castilla-la Mancha & Sicaman Nuevas Tecnologías Ciudad Real, Spain
Julio A. Sangüesa, University of Zaragoza, Spain
Carol Savill-Smith, Independent Researcher, UK
Oliver Schneider, DIPF - Deutsches Institut für Internationale Pädagogische Forschung / Hochschule Darmstadt, Germany
Ahmed Shahin, Zagazig University, Egypt
Roman Shtykh, Yahoo Japan Corporation, Japan
Mujdat Soyturk, Marmara University, Istanbul, Turkey

Copyright Information

Table of Contents

A Spectrum Sensing Technique Based on Variably Weighted Sensing Samples in the Presence of Random Traffic of Primary Users

Eunyoung Cho, Keunhong Chae, and Seokho Yoon
College of Information and Communication Engineering
Sungkyunkwan University
Suwon, South Korea
e-mail: syoon@skku.edu

***Abstract*—This paper investigates the spectrum sensing problem under the random traffic condition of the primary user (PU) in cognitive radio (CR) networks, where the PU may depart or arrive in a random way during the sensing period. Considering that the data transmission period of the secondary user (SU) starts right after the sensing period ends, we observe that, in the presence of the random traffic of the PU, the sensing samples in the latter part of the sensing period are more reliable in making a decision on whether the PU is present or not. Based on this observation, then, we propose spectrum sensing test statistics exploiting only sensing samples in the latter part the sensing period and assigning a larger weight to a sensing sample closer to the end of the sensing period. It is demonstrated in numerical results that the proposed methods offer a significant improvement in detection and receiver operating characteristic (ROC) performances over the conventional methods under the random traffic condition of the PU.**

***Keywords-** Cognitive radio (CR); Random traffic; Spectrum sensing; Primary user (PU).*

I. INTRODUCTION

With the explosive demands for various high date rate services in wireless communications, recently, the radio frequency spectrum has rapidly become a scarce resource, and thus, the cognitive radio (CR) has gained much interest with its capability of offering a high degree of efficiency in using the radio frequency spectrum [1]-[3]. Spectrum sensing is an essential task in CR, which detects a spectral hole of the frequency spectrum allocated to the primary user (PU), thus allowing the secondary user (SU) to share the frequency spectrum with the PU [4].

Conventionally, the spectrum sensing techniques [5]-[7] have been designed under the assumption that the status of the PU does not change during the sensing period (i.e., the PU is present or absent during the whole sensing time). However, it is clear that the status of the PU may change in a real environment, i.e., the PU may depart or arrive in a random way during the sensing period. Although several spectrum sensing techniques [8]-[10] have been presented with considering this random traffic of the PU, the techniques require the channel knowledge, such as the distributions of the departure and arrival times of the PU signal [8] and noise variance [9], or they employ the sensing samples in the initial part of the sensing period causing a wrong spectral hole detection with a high likelihood [10].

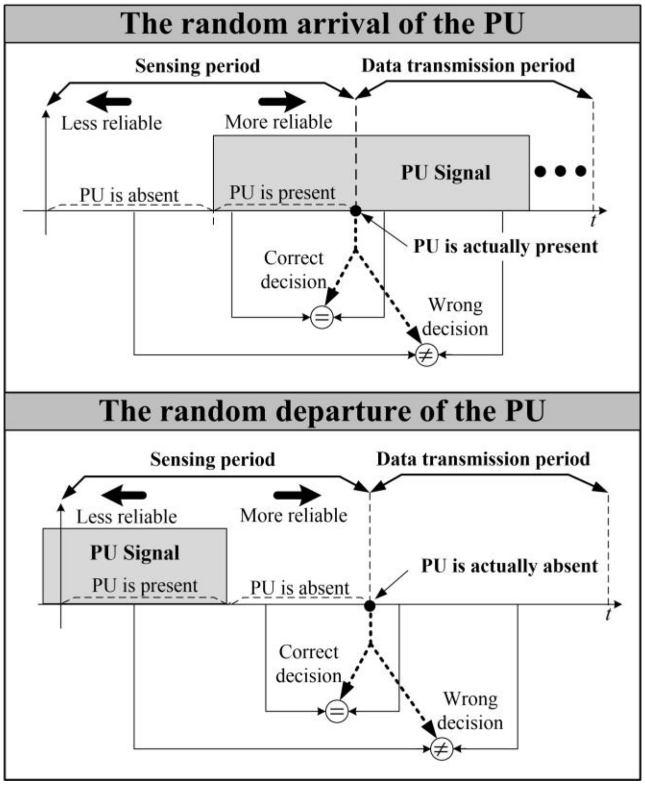


Figure 1. Spectrum sensing decision under the random traffic condition of the PU.

In this paper, we propose a spectrum sensing technique based on variably weighted sensing samples, where only sensing samples in the latter part of the sensing period are used in the spectral hole detection, and a larger weight is assigned to a sensing sample closer to the end of the sensing period (i.e., the sensing sample closest to the end of the sensing period has the largest weight). The proposed technique is expected to perform well in the presence of the random traffic of the PU signal, since a sensing sample closer to the end of the sensing period is more reliable in the decision on the presence and absence of the PU signal when the PU departs or arrives randomly during the sensing period, as shown in Figure 1 [11].

The remainder of this paper is organized as follows. Section 2 models the spectrum sensing problem under the random traffic condition of the PU as a binary hypothesis test. Section 3 describes the proposed technique. Section 4 compares the proposed and conventional techniques in terms of the detection probability and receiver operating characteristic (ROC) curve. Section 5 concludes this paper with a brief summary.

II. Random Traffic Model of the PU signal

The static and random traffic models of the PU signal are depicted in the left-hand and right-hand sides of Figure 2, respectively. In the static traffic model, the status of the PU signal remains unchanged during the whole sensing time, and thus, the spectrum sensing can be formulated as the following binary hypothesis testing problem [5]

$$H_0^s : z[i] = w[i] \qquad \text{for } i=1,2,\ \cdots\ ,I, \tag{1}$$

and

$$H_1^s : z[i] = s[i] + w[i] \quad \text{for } i=1,2,\ \cdots\ ,I, \tag{2}$$

where the hypotheses H_0^s and H_1^s represent the absence and presence of the PU signal during the whole sensing time, respectively, I is the total number of the sensing samples, and $z[i]$, $s[i]$, and $w[i]$ represent the ith samples of the received signal, the PU signal, and the additive noise, respectively.

In the random traffic model of the PU signal, on the other hand, the spectrum sensing is formulated as [8]

$$H_0^r : z[i] = \begin{cases} s[i] + w[i] & \text{for } i = 1, 2, \cdots, J_0, \\ w[i] & \text{for } i = J_0 + 1, J_0 + 2, \cdots, I, \end{cases} \tag{3}$$

and

$$H_1^r : z[i] = \begin{cases} w[i] & \text{for } n = 1, 2, \cdots, J_1, \\ s[i] + w[i] & \text{for } n = J_1 + 1, J_1 + 2, \cdots, I, \end{cases} \tag{4}$$

where the hypothesis H_0^r and H_1^r represent the absence and presence of the PU signal not during the whole sensing time but at the end of the sensing period, respectively, i.e., the PU signal is declared absent in the frequency band under consideration if it departs between the J_0th and $(J_0 + 1)$ th samples, and thus, is eventually absent at the I th sample instant, whereas the PU signal is declared present if it arrives between the J_1th and $(J_1 + 1)$ th samples and so is present at the I th sample instant. It is noteworthy that (3) and (4) reduce to (1) and (2), respectively, when $J_0 = J_1 = 0$.

III. Proposed Spectrum Sensing Technique

A. Test Statistics Based on Variably Weighted Sensing Samples

To obviate the need for the knowledge on the distributions of the departure and arrival times of the PU signal, we consider a spectrum sensing technique based on the energy detection [6] with the sensing samples, and also, to exclude the sensing samples in the initial part of the sensing period causing a wrong decision on the presence and absence of the PU signal with a high likelihood, we exploit only the last L samples out of the I samples. In addition, we enable the spectrum sensing technique to assign a larger weight to a sensing sample closer to the end of the sensing period, since a sensing sample closer to the end of the sensing period is more reliable in the decision on the presence and absence of the PU signal under the random traffic condition of the PU signal, as mentioned in Introduction. Bearing all of these desired features in mind, now, we propose the following two spectrum sensing test statistics

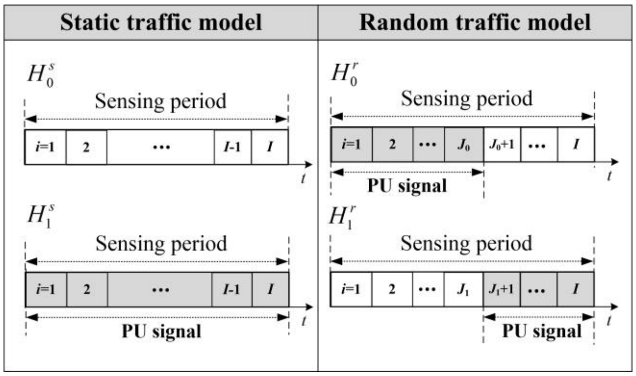


Figure 2. The static and random traffic models of the PU signal

$$T_P = \sum_{i=I-L+1}^{I} \left(\frac{i-(I-L)}{L} \right)^a |\,z[i]\,|^2, \tag{5}$$

and

$$T_E = \sum_{i=I-L+1}^{I} b^{\frac{i-(I-L)}{L}} |\,z[i]\,|^2, \tag{6}$$

where $a > 0$ and $b > 1$. The two test statistics are similar in that both of them assign a larger weight to a sensing sample closer to the end of the sensing period; yet, they are different in their weights: The weights of (5) and (6) are the power and exponential functions, respectively, of the normalized indices of the last L samples.

B. Distributions of Test Statistics

Assuming that the L noise samples $\{w[i]\}_{i=I-L+1}^{I}$ are statistically independent and identically distributed Gaussian random variables with zero mean and variance σ^2, we can derive the characteristic functions of the test statistics as

$$\Phi_0(j\eta) = \prod_{l=1}^{L} \frac{1}{\sqrt{1 - j2\eta\sigma^2\lambda_l}} \tag{7}$$

under H_0^r and

$$\Phi_1(j\eta) = \prod_{l=1}^{L} \frac{1}{\sqrt{1 - j2\eta\sigma^2\lambda_l}} \exp\left(\frac{j\eta s^2[I-L+l]}{(1 - j2\eta\sigma^2\lambda_l)} \right) \tag{8}$$

under H_1^r, where $\lambda_l = (l/L)^a$ and $b^{(l/L)}$ for T_P and T_E, respectively. The inverse Fourier transforms of (7) and (8) would yield the probability density functions (PDFs) under H_0^r and H_1^r, respectively; however, it is highly complicated to express the PDFs in a closed form. Noting that the values of a and b do not change the general forms of the PDFs, thus, we verify the validity of the characteristic functions by deriving the PDFs for a simplified case (i.e., when $a = 0$ and $b = 1$, and so, when $T_P = T_E$), and then, by comparing the analytical detection probability based on the PDFs with the simulated detection probability. The characteristic functions $\Phi_0(j\eta)$ and $\Phi_1(j\eta)$ reduce to

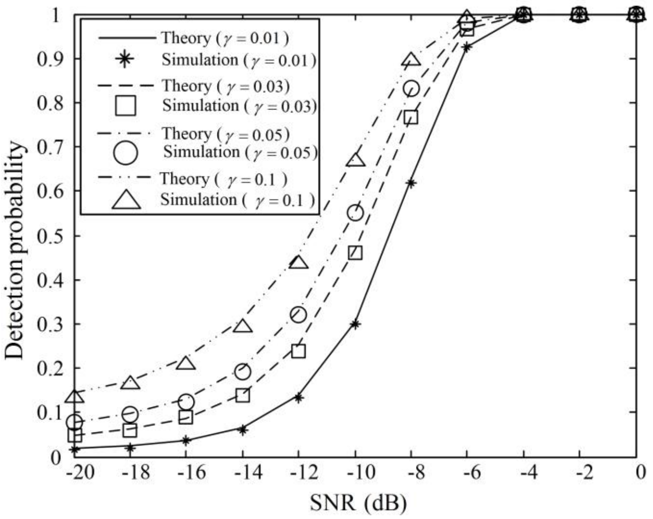


Figure 3. The analytical and simulated detection probabilities for T_P and T_E when γ =0.01, 0.03, 0.05, and 0.1 and L=50.

$$\frac{1}{(1-j2\eta\sigma^2)^{L/2}} \tag{9}$$

and

$$\frac{1}{(1-j2\eta\sigma^2)^{L/2}}\exp\left(\frac{j\eta\sum_{l=1}^{L}s^2[I-L+l]}{(1-j2\eta\sigma^2)}\right), \tag{10}$$

respectively, when $a=0$ and $b=1$. In fact, (9) and (10) are the characteristic functions of the central chi-square and non-central chi-square PDFs, respectively, with L degrees of freedom [12]. Thus, the detection probability is given by

$$\int_{\varepsilon}^{\infty}\left(\frac{1}{2\sigma^2}\right)\left(\frac{y}{\beta^2}\right)^{\frac{(L-2)}{4}} e^{\frac{-(\beta^2+y)}{2}} B_{L/2-1}\left(\frac{\beta\sqrt{y}}{\sigma^2}\right)dy \tag{11}$$

for both T_P and T_E, where

$$\beta^2 = \sum_{l=1}^{L} s^2[I - L + l], \tag{12}$$

$$B_{\alpha}(x) = \sum_{k=0}^{\infty}\frac{(x/2)^{\alpha+2k}}{k!\Gamma(\alpha+k+1)}, \tag{13}$$

is the α th-order modified Bessel function of the first kind with $\Gamma(x)=\int_0^\infty t^{x-1}e^{-t}dt$ and x>0, and ε is a threshold obtained from

$$\int_{\varepsilon}^{\infty}\frac{1}{\sigma^L 2^{L/2}\Gamma(\mathrm{L}/2)} y^{L/2-1}e^{-y/2\sigma^2}dy = \gamma \tag{14}$$

with γ a pre-determined false alarm probability. Figure 3 shows the analytical and simulated detection probabilities for T_P and T_E when γ =0.01, 0.03, 0.05, and 0.1 and L=50, where we can clearly see that the analytical and simulated results agree with each other, thus verifying the validity of (7) and (8), and allowing us to use them in determining a threshold for the detection performance evaluation in the next section.

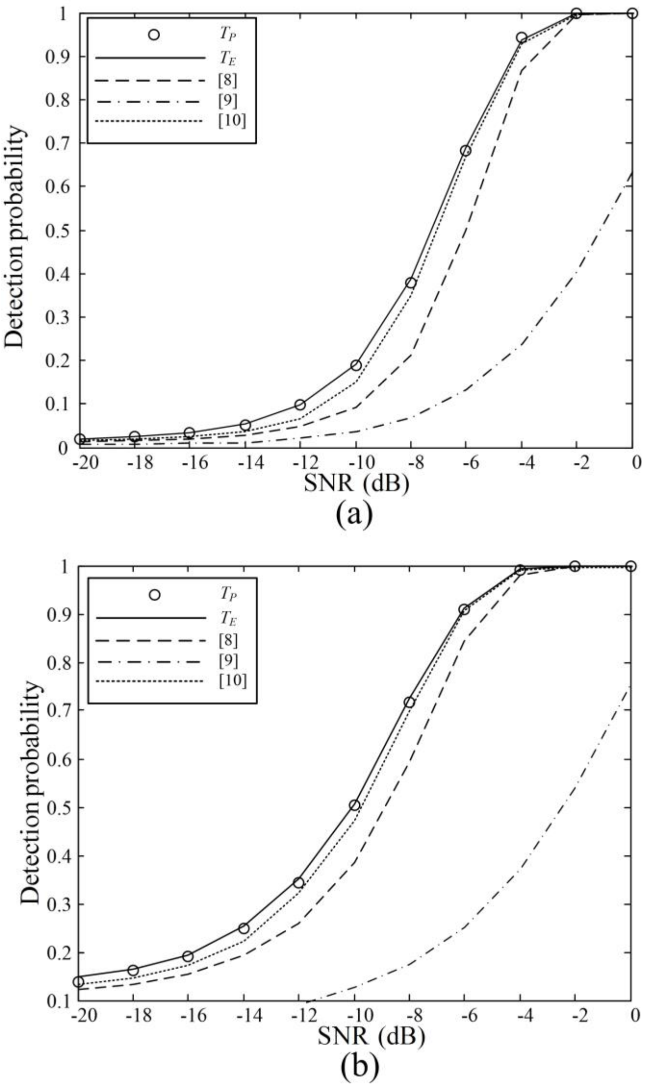


Figure 4. The detection probabilities as a function of the SNR of the proposed and conventional schemes when the false alarm probability is (a) 0.01 and (b) 0.1.

IV. Numerical Results

In this section, the proposed spectrum sensing technique is compared with the conventional spectrum sensing techniques in terms of the detection probability and ROC curve, where I is set to 200, J_0 and J_1 are assumed to be distributed uniformly over the sensing period, the signal-to-noise-ratio (SNR) is defined as $s^2[i]/E^2\{w[i]\}$ with $E\{\bullet\}$ denoting the statistical expectation, and L, a, and b are numerically optimized to maximize the detection probability for each of the given SNR values and false alarm probabilities.

TABLE I. THE OPTIMIZED VALUES OF $L, a,$ and b WHEN THE FALSE ALARM PROBABILITY IS 0.01 AND 0.1.

SNR (dB)			0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20
False Alarm Probability = 0.01	T_P	**Optimized *a***	0.4	0.3	0.7	1	1.6	2.2	2.7	2.4	2.6	2.2	2.3
		Optimized *L*	175	175	175	175	175	95	85	175	100	100	95
	T_E	**Optimized *b***	6.5	3	3.5	4.5	9.5	10.5	3	10.5	13	13	15
		Optimized *L*	175	175	175	175	150	125	65	95	105	125	110
False Alarm Probability = 0.1	T_P	**Optimized *a***	0.6	0.1	0.6	0.7	1.3	1.9	1.9	0.8	1.7	2.3	3
		Optimized *L*	150	150	175	175	175	175	150	95	110	150	90
	T_E	**Optimized b**	13.5	1.5	3	2.5	5	8.5	14.5	7.5	14.5	13.5	7
		Optimized *L*	175	150	175	150	175	150	175	150	175	80	150

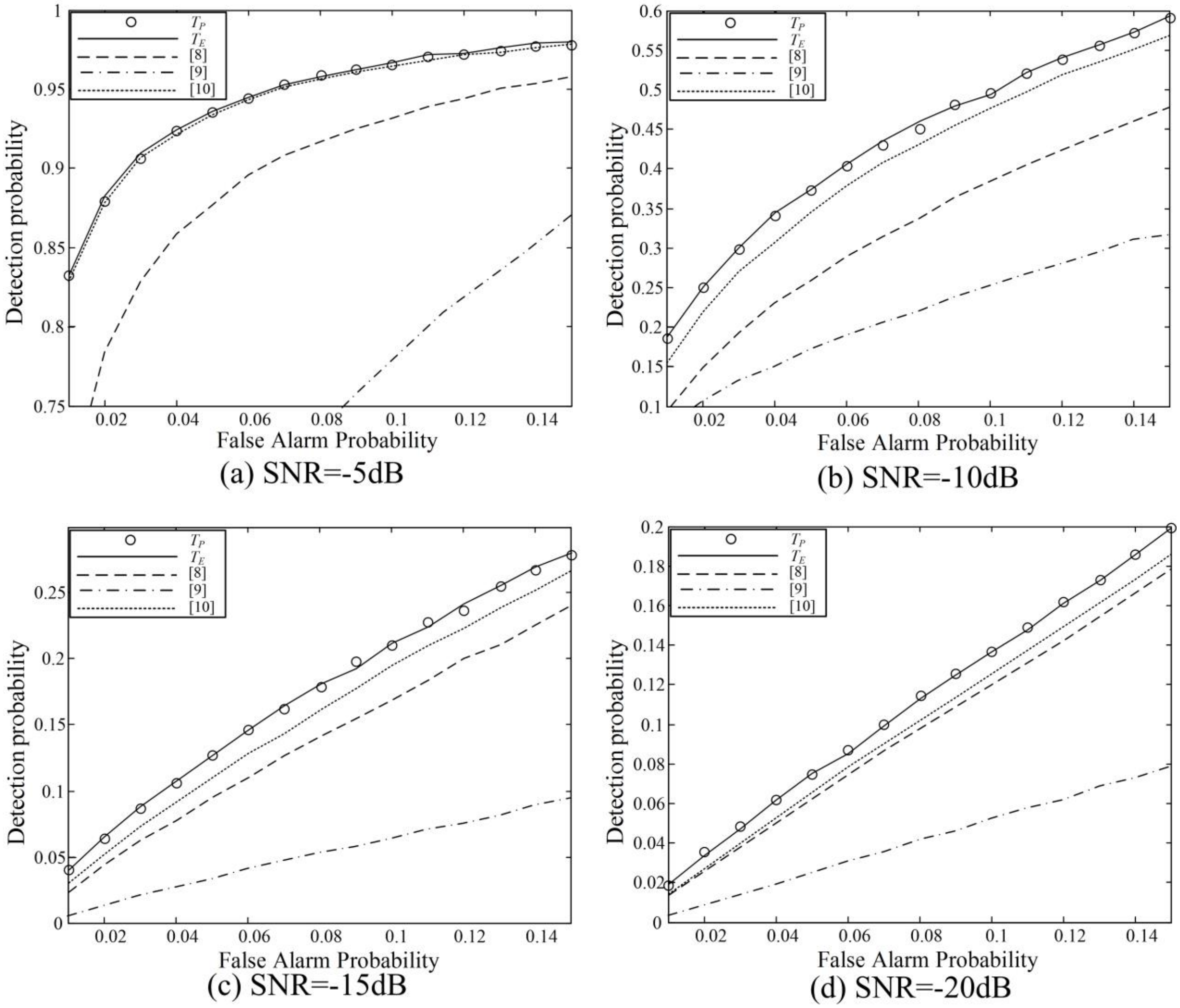


Figure 5. The ROC curves of the proposed and conventional techniques when the value of the SNR is (a) -5dB, (b) -10dB, (c) -15dB, and (d)-20dB.

Table I shows the optimized values of L, a, and b for various values of the SNR when the false alarm probability is 0.01 and 0.1, where it is observed that, as the value of the SNR decreases, the values of a and b generally increases to amplify the signal power, whereas the value of L generally decreases to exclude highly noise-contaminated sensing samples while preserving the reliable samples in the latter part of the sensing period.

Figure 4 shows the detection probabilities of the proposed and conventional spectrum sensing techniques as a function of the SNR when the false alarm probability is 0.01 and 0.1, where we can observe that the proposed techniques outperform the conventional techniques with a gain ranging approximately from 0.5 dB to 13 dB, which stems from the fact that the proposed techniques use only reliable sensing samples in the latter part of the sensing period unlike the conventional techniques. In addition, in the figure, we can see that the performance of T_E is slightly better than that of T_P, due to the effect of the weights on the sensing samples being slightly larger with T_E than with T_P.

Figure 5 shows the ROC curves of the proposed and conventional techniques when the value of the SNR is (a) -5dB, (b) -10dB, (c) -15dB, and (d) -20dB. It is seen in the figure that the proposed techniques offer an improvement in performance over the conventional techniques for all cases shown, and the improvement becomes more pronounced for a larger SNR value generally. This is because the reliable sensing samples are more efficiently utilized in the spectral hole detection through the proposed variable weighting methods, and the number L of the reliable sensing samples generally increases as the value of the SNR becomes larger, as shown in Table I.

Although the PU signal is assumed to arrive or depart only one time during the sensing period in this paper, the PU signal may arrive or depart several times [13] during the sensing period or even during the data transmission period [14]. So, we would like to address sensing techniques in such more realistic environments in the future work.

V. Conclusion

In this paper, we have proposed two novel detection test statistics based on variably weighted sensing samples for spectrum sensing under the random traffic condition of the PU signal. Using the power and exponential functions of the sensing samples in the latter part of the sensing period, we have designed weighting methods that enable the detection test statistics to assign a larger weight to a sensing sample closer to the end of the sensing period, and consequently, to improve their own decision reliability in the presence of the PU random traffic. Numerical results demonstrate that the proposed test statistics provide better detection and ROC performances than the conventional ones under the random traffic condition of the PU signal.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation (NRF) of Korea under Grant 2015R1D1A1A01057327 with funding from the Ministry of Education.

References

[1] S. Haykin, “Cognitive radio: Brain-empowered wireless communications,” *IEEE J. Select. Areas Comm.* vol. 23, no. 2, pp. 201-220, 2005.

[2] D. W. K. Ng, E. S. Lo, and R. Schober, “Multiobjective resource allocation for secure communication in cognitive radio networks with wireless information and power transfer,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3166-3184, 2016.

[3] S. W. Oh, Y. Ma, E. Peh, and M. H. Yao, TV white space: The first step towards better utilization of frequency spectrum, 1st ed.; John Wiley & Sons, 2016.

[4] A. Ghasemi and E. S. Sousa, “Spectrum sensing in cognitive radio networks: Requirements, challenges and design trade-offs,” *IEEE Communications Magazine*, vol. 46, no. 4, pp. 32-39, 2008.

[5] N. A. Hussien, E. Barka, M. Abdel-Hafez, and K. Shuaib, “Secure spectrum sensing in cognitive-radio-based smart grid using role-based delegation,” in *Proc. International Conference on Information Management and Engineering*, pp. 25-29, 2016.

[6] Y. Zeng and Y. C. Liang, “Eigenvalue-based spectrum sensing algorithms for cognitive radio,” *IEEE Transactions on Communications*, vol. 57, no. 6, pp. 1784-1793, 2009.

[7] H. Hu, H. Zhang, H. Yu, and Y. Chen, “Spectrum-energy-efficient sensing with novel frame structure in cognitive radio networks,” *AEU-International Journal of Electronics and Communications*, vol. 68, no. 11, pp. 1065-1072, 2014.

[8] N. C. Beaulieu and Y. Chen, “Improved energy detectors for cognitive radios with randomly arriving or departing primary users,” *IEEE Signal Process. Lett.*, vol. 17, no. 10, pp. 867-870, 2010.

[9] W. L. Chin, J. M. Li, and H. H. Chen, “Low-complexity energy detection for spectrum sensing with random arrivals of primary users,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 947-952, 2016.

[10] X. Xie, X. Hu, B. Ma, and T. Song, “Improved energy detector with weights for primary user status changes in cognitive radio networks,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, pp. 1-8, Article ID 836793, 2014.

[11] X. Xie and X. Hu, “Improved energy detector with weights for primary user status changes in cognitive radios networks,” in *Proc. Consumer Communications and Networking Conference*, pp. 53-58, 2014.

[12] J. G. Proakis, Digital communications, 4th ed.; McGraw-Hill, 2001.

[13] T. Düzenli and O. Akay, “A new spectrum sensing strategy for dynamic primary users in cognitive radio,” *IEEE Communications Letters*, vol. 20, no. 4, pp. 752-755, 2016.

[14] M. Amini, F. Hemati, and A. Mirzavandi, “Optimizing SU transmission time under collision constraint considering PU returns,” *IETE Journal of Research*, vol. 61, no. 6, pp. 679-685, 2015.

Energy-efficient Live Migration of I/O-intensive Virtual Network Services Across Distributed Cloud Infrastructures

Ngoc Khanh Truong*, Christian Pape*, Sebastian Rieger*, Sven Reißmann†
*Department of Applied Computer Science
Fulda University of Applied Sciences, Fulda, Germany
Email: {ngoc.k.truong, christian.pape, sebastian.rieger}@cs.hs-fulda.de
†Datacenter
Fulda University of Applied Sciences, Fulda, Germany
Email: sven.reissmann@rz.hs-fulda.de

***Abstract*—Virtual infrastructures and cloud services became more and more important over the past years. The abstraction from physical hardware offered by virtualization supports an increased energy efficiency, for example, due to higher utilization of underlying hardware through consolidation, or the ability to geographically move cloud services based on lowest available energy prices and renewable energy. This paper gives an overview on such migration techniques in distributed private cloud environments. The presented OpenStack-based testbed is used to measure migration costs along with the service quality of virtualized network services. The results can be used to evaluate whether network services and virtual resources can be migrated to distant sites to reduce energy costs. Correspondingly, the paper illustrates the impact of high memory and input/output (I/O) load on live migrations of network services and evaluates possible optimization techniques.**

Keywords–Cloud Computing; OpenStack; Network Services; Live Migration; Energy Efficiency.

I. Introduction

Energy costs are an important factor for data centers and IT infrastructures as a whole. Drivers for the increasing costs over the last years have been electricity prices, but also the growing energy demand of data centers and IT infrastructures. Regarding the electricity price, the changes in national energy policies to move from low-priced conventional, e.g., nuclear, power to renewable energies (e.g., in the European Union and especially in Germany), augur that energy costs will increase even further. While the percentage of the costs for network equipment and services have been negligible for data centers in the past, this is likely to change due to increased bandwidth and the steadily increasing number of network devices, amplified by the evolving ”Internet of Things” and cloud-based services. Recent papers even state that the network power consumption could grow beyond 25% [1][2] of the total data center energy demand. This is especially likely for large data centers (i.e., Google, Amazon, Facebook), whose inner data center traffic is quickly increasing [3]. Since virtualization is used for compute, storage and network resources in modern data centers, these infrastructures support automatic provisioning and management of virtual resources, that can be used to optimize the energy efficiency. For example, virtual resources can be consolidated to reduce the required hardware based on the current load. During off-peak hours, resources and links can be powered down or use power management, while being quickly and automatically reactivated on demand. This also allows for elastic scalability [4], as well as adaptive scheduling, placement and migration of virtual resources. The scheduler can consider electricity prices and the availability of renewable energy resources across multiple data centers [5]. Hence, an energy- and cost-efficient adaptive placement of virtual resources can be attained. Nevertheless, network services impose special requirements for live migrations. The network load, e.g., on virtual network functions (VNF), is typically higher than on back end servers, due to their function as a front end for multiple services or servers. This leads to a high I/O rate of the virtual machines (VMs) and containers offering such virtual network services (e.g., VNF). Sometimes, these I/O-intensive memory and network operations are enhanced by using special acceleration functions of the underlying hardware, i.e., TCP offloading or single-root I/O virtualization (SR-IOV), that also hold specific constraints for live migrations.

In this paper, an analysis of the impact of these implications for the migration of virtual network services across distributed cloud environments is presented. Our approach uses an OpenStack-based testbed migrating virtual network services under load and evaluating the results. Additionally, techniques to improve the energy efficiency of the migration are discussed. By using a live migration, the services can be transferred seamlessly during operation instead of interrupting existing connections leading to additional energy being required to reestablish lost connections. However, the energy consumption of the migration itself needs to be optimized (e.g., limiting resources and time needed for the migration).

The rest of this paper is laid out as follows. Section II presents related work and defines the research questions of this paper. In Section III, the state of the art in energy-efficient private clouds, as well as the usage of virtual network services and live migration of virtual resources in such infrastructures are described. The model for our approach is introduced in Section IV, describing the requirements for scheduling and migrations of virtual network services in private clouds, to support an energy-efficient placement. Section V characterizes the testbed that we created to measure the impact of virtual network service migrations on the energy efficiency of private cloud infrastructures and presents the results of the evaluation. Finally, Section VI draws a conclusion, discusses the findings of the evaluation compared to the related work, and gives an outlook on further research that we will carry out in this area.

II. Related Work

Migration of virtual resources and its impact on application performance is subject of current research. The energy-efficient placement of virtual machines in an OpenStack-based environment is discussed in [6][7]. Indeed, these approaches target on the algorithms used for placing VMs based on temperature and cooling demands, but also focus on network requirements for the VMs. A vector-based algorithm for virtual machine placement considering the availability of renewable energy is discussed in [8]. Furthermore, more general evaluations are given in [9] and [10]. These publications examine the relevant parameters for an energy-efficient placement of VMs in a data center. A basic analysis of VM migration costs and the impact of migration on application performances is discussed in [11]. In [12] an estimation of the energy consumption of physical servers running VMs and an algorithm for energy-efficient VM placement are described. The well-known ElasticTree project [13] focusses on energy-efficient computer networks by throttling network components using OpenFlow. Other projects like ECODANE [14] extend these ideas to also provide traffic-engineering techniques. Constraints and requirements for energy-efficient placement of VMs related to their network connectivity were introduced in [15][16][17]. An evaluation of the power consumption during VM migration tasks is presented in [5]. This publication also includes a breakdown on different data center components like storage, network and compute resources. Furthermore, [18] discusses an energy-aware virtual data center architecture using software defined networks (SDN). Finally, [19] introduces benchmarking test metrics for performance and reliability monitoring and discusses related issues. A study comparing different hypervisors concerning migration time and efficiency is presented in [20]. The interference effects of simultaneously running migrations and the efficiency of different permutations of migrations are reviewed in [21].

III. State Of The Art

The evolution of cloud services in IT infrastructures enables companies to speed up business processes and scale their services on demand. Physical servers, storage and network devices are consuming energy, but today these components are typically just the foundation for virtualized workload running on top of them. Furthermore, in such highly virtualized environments, the virtual resources providing the services are the consumers of power and bandwidth. Orchestration and automation techniques like SDN can help to optimize the power consumption in cloud infrastructures. To ensure the service quality and scalability along with the energy efficiency, it is necessary to investigate the behavior of these virtual resources, e.g., regarding available migration techniques.

A. Energy-efficient Private Clouds

Today, energy efficiency and power management is a foundation pillar in modern data centers. This is mainly driven by increasingly high energy costs and energy consumption in large-scale IT infrastructures. Data centers are using a large amount of power not only for running the IT components and equipment, but also for cooling them. The ratio between energy consumed by IT equipment and the overall power consumption including cooling and energy loss in power supplies is known as the power usage effectiveness (PUE). This value describes the operational overhead of data centers and is an eligible candidate for optimization approaches.

The concept of cloud computing enables companies to better utilize their physical IT resources and empowers them to dynamically scale their services in a location-independent manner. To take advantage of these benefits, a consequent resource management must be deployed. Ideally, this means that currently not required compute resources, as well as their dependencies like upstream or downstream storage or network devices are partially or fully suspended or shut down. The consumption of energy in a common cloud environment depends on its directly associated physical infrastructure components like compute resources (i.e., central processing unit - CPU, random access memory - RAM), storage devices (i.e., storage area networks - SAN, network attached storage - NAS, local or direct-attached storage) and network components (i.e., routers, switches, firewalls). Thus, the power consumption of a service depends on the physical IT resources that are needed to provide it. However, VMs providing cloud services are not picky concerning their location of execution, as long as required dependencies are met at either site.

By migrating virtual resources across distant data centers in different regions, it is possible to optimize energy efficiency and cost. Such ”follow-the-sun” data center services move their workload to different geographic regions to more efficiently balance computing demand while taking into account the latency for the end users to access the service. Usually, the output of renewable energy sources is fluctuating, which means that the energy is not always available when needed and also not necessarily produced near the point where it is consumed. Further, energy storage at industrial scale is not available yet. Related to that, this also leads to seasonal and regional energy price fluctuations. The cloud paradigm enables companies to move their workload nearby the currently available renewable energy sources and to take advantage of the economic benefits by consuming energy at lower prices.

B. Migration of Virtual Resources in Private Clouds

Today’s cloud software is providing a layer for scalable and elastic cloud applications that allows to deploy virtual network services (e.g., VNFs) like routers, load balancers or firewalls. Also, private cloud platforms like OpenStack already added a lot of these functions to their service portfolio. As a result, many industry-leading service providers are starting to use OpenStack as a platform to deliver reliable and scalable services and applications. This includes VMs running customer-facing applications, as well as virtualized storage and networking components needed for the service delivery. Of course, containers as a very thrifty and scalable building block for cloud services can also be provisioned and deployed in these infrastructures. However, to offer reliable, elastic and energy-efficient services, these resources have to be movable across the infrastructure components. This movability of virtual resources is mostly provided by VM migration from one node to another. The migration can be implemented live or online by transferring block storage of the VM or using a shared storage back end, and finally transmitting the main memory and CPU state. Furthermore, a VM can also be migrated offline by suspending, transmitting its state and resuming the machine consecutively. These approaches are described in detail in Section IV-B. When a VM does not contain any essential data and the configuration can be realized by an automated provisioning mechanism, it is also possible to just destroy a VM or container on the source node and recreate or respawn it on the destination node. It is obvious, that this technique minimizes network transfer costs and requirements for shared storage hardware but also implies that the cloud application or service is well-designed related to elasticity. Moreover, live migration techniques for containers are currently developed and discussed. While the small size of containers compared to VMs reduces the network traffic for the migration, saving the state of containers holds much more dependencies and hence is more difficult to implement [22].

IV. Energy-efficient Placement of Virtual Network Services in Private Clouds

The migration of virtual network services to regions where renewable energy sources are currently available or where energy prices are lower, can substantially improve the overall energy efficiency. However, if the costs for the migration are too high, e.g., due to a reduced performance of the migrated resources, the migration will be inefficient. For these reasons, when designing services, it is important to understand how the migration process is performed in the underlying infrastructure to restrict the consequences of migration costs.

A. Scheduling

A common OpenStack environment is based on multiple services handling different aspects of the cloud environment. First of all, Nova, the compute fabric controller, encapsulates the hypervisor and is responsible for the execution of VMs. Block-level storage is provided by the Cinder service. It manages the complete life cycle of block devices for the virtual servers. The image service Glance stores disk and server images and their metadata and assures, that they are available to the compute nodes. The networking component Neutron manages multi-tenant virtual networks supporting different network architectures. Also, OpenStack Neutron already offers some virtual network services (i.e., VNF) like firewalls and load balancers as a service. While OpenStack contains additional components, this paper is based on the OpenStack core services described above. Scheduling and placement of virtual resources in OpenStack environments is carried out by schedulers of the services given above. For example, the nova-scheduler checks which compute nodes can provide the requested resources. The decision is based on filters (i.e., based on capacity, consolidation ratio, affinity groups) that can be modified by an administrator.

B. Migration Techniques

One of the crucial points when performing the migration is to ensure that services should not be disrupted during the migration process, otherwise possible service-level agreements (SLAs) will be violated. OpenStack, which typically uses libvirt and the kernel-based virtual machine (KVM) hypervisor, provides three different migration types to move VMs from the source host to a destination host with almost no downtime: shared storage-based live migration, block live migration and volume-backed live migration [23]. Shared storage-based live migration, as the name states, requires a shared storage that is accessible from source and destination hypervisors. During the migration only the memory content and system state (e.g., CPU state, registers) of the VM are transferred to the destination host. This migration type in OpenStack can be performed using a pre-copy [24] or post-copy [25] approach. In the former, VM memory pages are iteratively copied to the target without stopping the services running on the migrated VM. Every change on memory state (i.e., dirtied memory) during the copy phase will trigger another transfer of modified memory pages. If predefined thresholds have been reached, e.g., the number of iterative copy rounds or the total amount of transmitted memory, or the amount of modified memory pages in the preceding copy round is small enough [26], the copy process is terminated, whereby the source VM is suspended, the source hypervisor copies the remaining modified memory pages and system state and resumes the VM on the destination. Depending on the dirtied page rate this switching can cause a downtime. A big issue of pre-copy migration arises at the iterative copy rounds. If the rate of memory change exceeds the transferred rate over the network, then the copy process will run infinitely. This limit can be eliminated by post-copy migration, in which at the beginning of the migration the migrating VM is stopped on the original node, then the non-memory VM state is copied to the destination, after which the VM will be resumed on the target. In parallel, a prepaging will be performed. At this stage, the memory pages are proactively pushed by the source to the destination VM. Any access to the memory pages on the target VM that have not yet been copied, result in the generation of page faults, requiring to transfer the accessed memory pages over the network. This process is known as demand paging. Obviously, this behavior can solve the indefinitely migration problem, but can cause a huge degradation of VM performance because of the large amount of page faults transferred over the high-latency medium in comparison to pre-copy migration. Moreover, post-copy cannot recover the memory state of the migrated VM in the case of network failure during the transfer of the page faults.

As the requirement of a shared storage increases the financial burden, block live migration is considered more cost effective. No shared storage is required when the migration takes place. Hence, this migration type is especially useful when moving the VMs between two sites over long distances without having to expose their storage to one another. This type is very similar to Microsoft Hyper-V Shared-Nothing Live Migration feature [27]. Initially, not only a VM on the remote host is created, but also the virtual hard disk on the remote storage. During the migration, at first the virtual hard disk contents of the running VM must be copied to the target host. Changes of disk contents as a result of write operations will be synchronized to the destination hard disk over the network. After the migration of the VMs storage is complete, the copy rounds of memory pages are executed which perform the same processes used for shared storage-based live migration. Once this stage is successfully finished, the target hypervisor will resume the VM, while the source hypervisor deletes the VM and its associated storage. Volume-backed live migration behaves like shared storage-based live migration since VMs are booted from volumes provisioned by Cinder instead of ephemeral disk, i.e., VM disks on shared storage. To achieve energy-efficient placement of VMs, the migration costs must be taken into account. These costs play an important role for the scheduling process to decide when and how often services should be migrated to remote hosts.

Two categories of parameters to calculate migration costs will be analyzed in this paper: total migration time, which denotes how long the migration lasts from the start of copy rounds until the VM is resumed on the remote host, and performance loss, which focuses on the degradation of the services performance during the migration process. Apparently, these costs are strongly impacted by the iterative copy rounds due to any modification on memory pages or disk contents. They should be thoroughly calculated to allow the scheduler to efficiently place services not only in terms of energy, but also their quality of service.

V. Evaluation

This experimental study concentrates on the impact of migration on memory- and I/O-intensive services. For this purpose, we set up an experiment in an OpenStack environment that is presented in the following sections.

A. Testbed Environment and Methodology

Our testbed environment consists of two physical servers that act as compute nodes and two NetApp E2700 providing block storage over 16 Gbit/s FibreChannel. Each of the compute nodes running Ubuntu 14.04 is equipped with two 8-core Intel(R) Xeon(R) E5-2650v2 2.60 GHz CPUs and 256GB of main memory. The nodes are connected using two 1 Gbit/s Ethernet interfaces over a Cisco C3750 switch. All migrated VMs run Ubuntu 14.04 with 1 vCPU, 2 GB of memory and 10 GB of disk space. In our study, migration costs of a web proxy as a virtual network service is analyzed. 10 VMs (Set 1) representing web proxy servers are initially launched on Nova-Compute 1 with a defined memory workload using the tool *stress*, which keeps dirtying the predefined amount of memory. We also activated swapping to simulate additional I/O load on the service. If all memory for user space (1702 MB, 83% of memory size) is already allocated, inactive memory pages will be swapped out to disk.

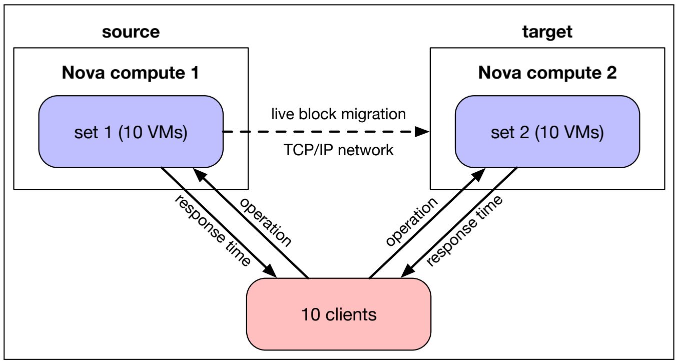


Figure 1. Overview of the methodology of the experiment.

The performance of each VM will be measured by 10 clients, each sending HTTP requests to the VMs in a fixed time interval. Additionally, we produce extra load on those VMs by sending other requests for various operations from the clients, such as searching a directory, writing a 20 MB file (disk I/O load) and generating 4096 bit RSA (stands for R. Rivest, A. Shamir and L. Adleman, see [28]) keys (CPU load). The response times for those requests are then used as a performance metric. After 15 minutes of measurement the same process is performed on 10 VMs of Set 2 on Nova-Compute 2. All source VMs are then concurrently migrated from Nova-Compute 1 to Nova-Compute 2 using block live migration. We chose block live migration due to its advantage in the case of moving the VMs located on two sites with large distance. While also 10 GBit/s Ethernet is available in our servers and switches, we used the 1 GBit/s NICs to better reinforce small effects of different migration parameters and changes. Furthermore, we varied the number of concurrent migrations to better understand the impact of the bandwidth on the migration. The performance of VMs on Nova-Compute 2 was also investigated to observe the influence of the migration on instances running on the target host. Figure 1 shows an overview of the methodology.

Besides several configurations that were necessary to implement a true live migration in OpenStack [23], the *max_requests* and *max_client_requests* parameters in libvirt had to be increased to 40, to support the large number of 10 concurrent migrations in the experiment. The experiment was performed using a script and was repeated 10 times. After changing a parameter in the experiment (e.g., the memory workload shown in Figure 2) it was run 10 times again. All runs led to reproducible results.

B. Research Results and Discussion

Figure 2 demonstrates the experimental results for different memory workloads. The results show, that the total migration downtime increases proportionally with stressed memory size caused by the iterative transfer of dirtied memory pages generated by the command-line tool *stress*. Another reason for this effect is the more intensive swapping of memory pages leading to a repeated modification of disk contents and thus more additional transfers over the network. In addition, the block live migration process in OpenStack will last longer, if we reduce the number of VMs migrated concurrently. The source of this impact is the overhead of nova-scheduler handling the migration requests.

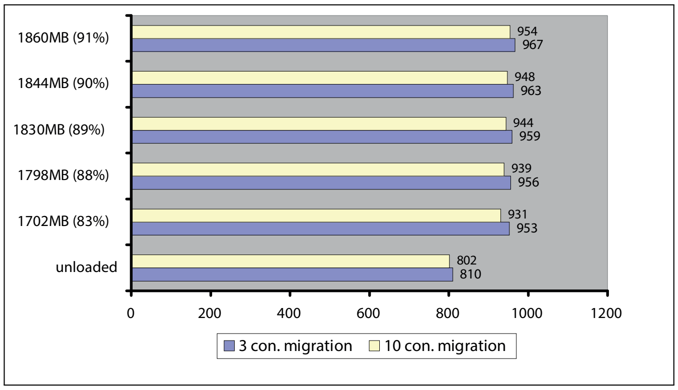


Figure 2. Total migration time (in seconds).

During the migration process, we observed that the performance for search operations within the VMs degrades significantly starting from 1830 MB loaded-memory (89% of total memory size). This degradation is shown in Figure 3, which demonstrates the response time for search operations on both sets before, during and after concurrently migrating 10 VMs of Set 1 to Nova-Compute 2. Response times were capped to a maximum of 60 seconds as seen in the figure for the second set before its creation. The average response time on Set 1 during the copy rounds rises from 2.299s to 5.606s, approximately 144%. Moreover, the migration of Set 1 to Nova-Compute 2 influences the VMs performance for search operations on this node. Particularly, the average search response time of Set 2 increases around 110% from 2.45s to 5.164s. After the VMs are moved to Nova-Compute 2, the performance of both sets is also decreased, by approximately 72% on Set 1 and 61% on Set 2, since Set 1 produces more I/O workload on the disk of the target host. The peak in Figure

3 during the migration denotes the switch process that was explained in Section IV-B.

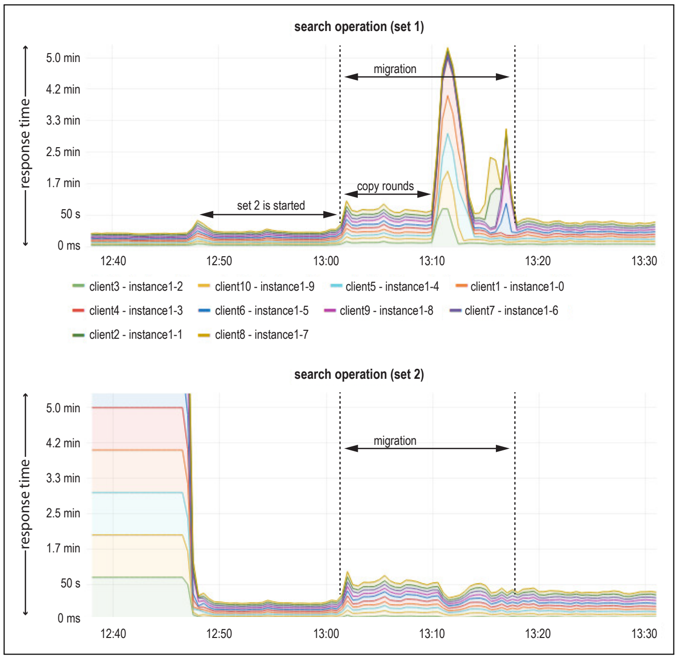


Figure 3. Performance of search operations with a memory workload of 1830MB on Set 1 and Set 2 before, during and after the migration.

Another conspicuous point is that the performance loss during the migration strongly depends on the amount of stressed memory as shown in Table I. The performance loss increases linear with the size of the memory workload. This could be due to the fact that the available amount of memory for buffer/cache used for I/O operations is too low so that more intensive I/O flush processes occur. Consequently, more disk synchronization must be performed over the network during the migration, causing a slowdown in the response times. In Figure 3, we can recognize that the performance of Set 1 for search operation slightly degrades when the VMs of Set 2 on Nova-Compute 2 are started, although they do not use a shared storage. For instance, the average response time of Set 1 increases from 2.067s to 2.532s (22.5%) in the case of 1830 MB loaded-memory, from 2.229s to 3.083s (39.7%) in the case of 1844 MB loaded-memory and from 2.856s to 6.858s (140%) in the case of 1860 MB loaded-memory. This result shows that many simultaneous intensive I/O operations on an extremely memory-intensive VM have an immense impact on the I/O performance of the underlying system in OpenStack and on the performance of I/O operations in hosted VMs, respectively. Nevertheless, this effect does not emerge if the stressed memory falls below 1830 MB, as well as for other non-I/O-related operations.

TABLE I. PERFORMANCE LOSS OF SEARCH OPERATION WITH DIFFERENT MEMORY WORKLOADS.

VM set	*Increased response time during migration (s)*			*Increased response time after migration (s)*		
	1830M	1844M	1860M	1830M	1844M	1860M
Set 1	3.307	4.389	6.241	1.655	2.527	3.431
Set 2	2.712	3.678	3.422	1.498	2.044	1.265

Last but not least, the performance of the main operation of the web proxy, serving HTTP requests, as well as the performance of the CPU-related operation, generating a 4096 bit RSA key, are only significantly impacted as the amount of stressed memory rises above 1860 MB. The average response time for HTTP requests to the migrating set grows from 0.166s to 0.785s during the migration process, whereas the one for the operation of generating an RSA key rises from 3.759s to 6.3s. This degradation effect arises only if those operations are carried out while other I/O-intensive operations such as a search for a file are running. When we perform block live migration with separate operations, the performance deviation did not occur. Therefore, it could be stated that not only I/O operations are strongly impacted by the migration process, but also have direct influence on the other operation types.

VI. CONCLUSIONS AND FUTURE WORK

Energy costs are an important factor for today's IT infrastructures, due to rising energy prices and increasing power consumption. The virtualization offered for compute, storage and network resources, e.g., in private clouds, allows for a seamless and transparent migration of virtual resources due to the abstraction from the underlying hardware. These migration techniques can be used to enhance the energy efficiency in data centers and have been constantly evolving over the last years. This includes adaptive migration, e.g., to consolidate or enhance the utilization of physical resources, as well as long-distance migration, that is not only covered by the related work and research presented in this paper, but also by current virtualization and hypervisor products (e.g., the introduction of long-distance vMotion in VMware vSphere 6 that was previously already available in Microsofts Hyper-V). Regarding the energy efficiency, however, additional costs of the migration itself have to be taken into account. These costs can either directly (i.e., higher load on the physical compute, storage and network resources) or indirectly gain energy costs, e.g., if the migrated services and applications cannot provide the same service quality during the migration. Hence, to improve the energy efficiency by using live migration techniques offered in cloud infrastructures, the migration costs need to be minimized. This especially holds true, if the migration is used to benefit from lower energy prices or the availability of renewable energy at distant data center sites.

Based on our previous research projects in this area, in this paper we present an evaluation of the migration costs for I/O-intensive VMs in an OpenStack environment. Due to the incoming and outgoing network traffic, especially virtual network services operated in VMs typically have a large I/O footprint in the infrastructure that is typically compensated by using hardware acceleration (i.e., virtual switch or kernel enhancements, data plane development kit - DPDK, SR-IOV). To be able to measure the additional load caused by a live migration of such services, and to quantify the impact on the service quality, we used additional tools (i.e., *stress*, *openssl*, *dd*, *find*) to add artificial I/O load on the machines while migrating them to another physical host in the OpenStack infrastructure. Based on the findings presented in this paper, the migration time increases proportionally to the added artificial I/O load. Furthermore, the load on storage and network resources grows accordingly as expected. The burden of the ongoing live migration can especially be measured if more than 80% of the total memory of the VM are continuously utilizes and changed. Interestingly, the migration time can be reduced by increasing the number of concurrent live migrations. This is due to the impact of the scheduler and message bus, handling the migrations in OpenStack together with libvirt and KVM. Similar effects can be observed with other hypervisors like vSphere or Hyper-V, though these products typically limit the number of parallel live migrations to small values.

The results of the experiments show a significant performance decrease for I/O read operations on the VMs during the migration. This conspicuous effect is likely due to limited available buffer/cache and extensive flush operations during the migration. The impact on the underlying OpenStack infrastructure leveraging libvirt and KVM, can also be observed in a performance decrease during start of VMs with high I/O and memory load, even if the VMs are running on separate hosts using different block storage. Several I/O operations (i.e., using *dd*, *find*, *stress*) were used to evaluate this decrease while constantly monitoring the service quality of the main operation. During the migration, a *find* process across the files on the VMs experienced a significant performance decrease. Also, VMs running on the target machine for the migration, experience a significantly reduced performance during this period. Moreover, for high additional artificial I/O loads, the main operation of the virtual network service was also impacted accordingly. Response times on the proxy increased from 0.166s to 0.785s during the migration. The high I/O load on the VMs leads expectedly to higher overall response times as more and more VMs are consolidated on a single physical host. However, a previous paper [5] presented an expected increase of the overall energy efficiency due to the higher utilization of the physical host, made possible by this consolidation.

Building on the results presented in this paper, we are currently focusing our research on live migration techniques for containers as a lightweight virtualization alternative compared to full-size VMs. Some types of services allow migration and scaling by simply destroying the containers at one site and respawning them at another. The required live migration techniques for containers are still being developed (e.g., in CRIU [22]) and are also within the focus of some related research projects. Initial results of our experiments show that the transferred amount of data during container migrations is expectedly less compared to VMs. Conversely, the migration process itself is more difficult, as the entire state of a process stack in the operating system needs to be stored and transferred. Existing checkpoint and restore techniques need to be extended to support live migration of container-based virtual network services. As virtualization techniques like containers are evolving, the requirement to seamlessly migrate virtual resources is likely to grow.

REFERENCES

[1] Greenberg, A, Hamilton, J, Maltz, D A, and Patel, P, "The cost of a cloud: research problems in data center networks," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, Dec. 2008, pp. 68–73.

[2] T. Cheocherngngarn, J. H. Andrian, D. Pan, and K. Kengskool, "Power efficiency in energy-aware data center network," in Proceedings of the Mid-South Annual Engineering and Sciences Conference, 2012.

[3] A. Andreyev, "Introducing data center fabric, the next-generation Facebook data center network," 2014, URL: https://code.facebook.com/posts/, 2017.06.07.

[4] P. Mell and T. Grance, The NIST definition of cloud computing. Washington DC: National Institute of Standards and Technology, 2011, URL: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, 2017.07.31.

[5] K. Spindler, S. Reissmann, and S. Rieger, "Enhancing the energy efficiency in enterprise clouds using compute and network power management functions," in ICIW 2014, The Ninth International Conference on Internet and Web Applications and Services, 2014, pp. 134–139.

[6] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing. IEEE Computer Society, 2010, pp. 826–831.

[7] ——, "Openstack neat: A framework for dynamic consolidation of virtual machines in openstack clouds–a blueprint," Cloud Computing and Distributed Systems (CLOUDS) Laboratory, 2012.

[8] C. Pape, S. Rieger, and H. Richter, "Leveraging Renewable Energies in Distributed Private Clouds," MATEC Web of Conferences, vol. 68, Aug. 2016, p. 14008.

[9] A. Song, W. Fan, W. Wang, J. Luo, and Y. Mo, "Multi-objective virtual machine selection for migrating in virtualized data centers," in Pervasive Computing and the Networked World. Springer, 2013, pp. 426–438.

[10] N. A. Singh and M. Hemalatha, "Reduce Energy Consumption through Virtual Machine Placement in Cloud Data Centre," in Mining Intelligence and Knowledge Exploration. Springer, 2013, pp. 466–474.

[11] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware. Springer-Verlag New York, Inc., 2008, pp. 243–264.

[12] D. Versick, D and D. Tavangarian, "CAESARA-combined architecture for energy saving by auto-adaptive resource allocation." in DFN-Forum Kommunikationstechnologien, 2013, pp. 31–40.

[13] B. Heller et al., "ElasticTree - Saving Energy in Data Center Networks." NSDI, 2010.

[14] T. Huong et al., "Ecodanereducing energy consumption in data center networks based on traffic engineering," in 11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop Visions of Future Generation Networks (EuroView2011), 2011.

[15] V. Mann, K. Avinash, P. Dutta, and S. Kalyanaraman, "VMFlow: Leveraging VM Mobility to Reduce Network Power Costs in Data Centers." Networking, vol. 6640, no. Chapter 16, 2011, pp. 198–211.

[16] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," Computer Networks, vol. 57, no. 1, 2013, pp. 179–196.

[17] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "Carpo: Correlation-aware power optimization in data center networks," in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1125–1133.

[18] Y. Han, J. Li, J. Y. Chung, J.-H. Yoo,, and J. W.-K. Hong, "SAVE: Energy-aware Virtual Data Center embedding and Traffic Engineering using SDN." NetSoft, 2015, pp. 1–9.

[19] T. Kim, T. Koo, and E. Paik, "SDN and NFV benchmarking for performance and reliability." APNOMS, 2015, pp. 600–603.

[20] W. Hu et el., "A quantitative study of virtual machine live migration." CAC, 2013.

[21] K. Rybina, A. Patni, and A. Schill, "Analysing the Migration Time of Live Migration of Multiple Virtual Machines." CLOSER, 2014.

[22] K. Kolyshkin, "Criu: Time and space travel for linux containers," 2015, URL: http://de.slideshare.net/kolyshkin/criu-time-and-space-travel-for-linux-containers, 2017.06.07.

[23] O. Found., "Openstack administration guide," 2017, URL: http://docs.openstack.org/admin-guide-cloud/index.html, 2017.06.07.

[24] C. Clark et al., "Live Migration of Virtual Machines." NSDI, 2005.

[25] M. R. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," ACM SIGOPS Operating Systems Review, vol. 43, no. 3, 2009, p. 14.

[26] A. Strunk, "Costs of virtual machine live migration: A survey," 2012 IEEE Eighth World Congress on Services, 2012, pp. 323–329.

[27] Microsoft, "Virtual machine live migration overview," 2015, URL: https://technet.microsoft.com/en-US/library/hh831435.aspx, 2017.06.07.

[28] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." Commun. ACM, vol. 21, no. 2, 1978, pp. 120–126.

An Over the Air Update Mechanism for ESP8266 Microcontrollers

Dustin Frisch[*], Sven Reißmann[†], Christian Pape[*]
[*]Department of Applied Computer Science
Fulda University of Applied Sciences, Fulda, Germany
Email: {dustin.frisch, christian.pape}@cs.hs-fulda.de
[†]Datacenter
Fulda University of Applied Sciences, Fulda, Germany
Email: sven.reissmann@rz.hs-fulda.de

***Abstract*—Over the last years, a rapidly growing number of IoT devices is found on the market, especially in the area of the so-called smart home. These devices, which are deployed in vast numbers, are frequently in use over many years. They pose a risk to the users privacy and to the internet as a whole if not provided regularly with security patches. Hence, a fully automated process for large-scale software updates of such embedded devices must be considered. In this article, we present an implementation of a durable and stable system for building and publishing cryptographically secure firmware updates for embedded devices based on *ESP8266* microcontrollers. This include mechanisms to build the updates from source and automatically sign, distribute and install them on the target devices.**

Keywords–IoT; Secure Updates; Over the Air; ESP8266.

I. Introduction

In todays marketplace, an explosive growth can be observed in the area of so-called smart devices, often referred to as Internet of Things (IoT). Conventional devices (e.g., door locks, light bulbs, washing machines) are extended with smart functions for remote control and monitoring. To implement the additional smart functions, small embedded computer systems are getting integrated into the devices, allowing them to connect to the local WiFi network.

In embedded systems, the software, also known as firmware, is an essential part of the system. On one side, it interacts with the hardware in a system specific way by implementing the specifications required by the components used in the system. On the other side, it provides use-case dependent functionality in interaction with general purpose hardware components. Embedded systems are often thought as systems that never change their requirements or functionality. However, practical use shows that the environment in which these systems run does, in fact, change. These changes include, but are not limited to, modifications to the expected behavior or additions to it, reconfiguration of parameters related to the communication with other systems or the users, as well as correcting errors, particularly security related issues, that have been reported after deployment and roll-out. In almost all cases, the requirements can be accomplished by changing the firmware and do not need any modification to the hardware. For updating the firmware on a system being deployed, the system must provide an interface for altering the firmware. In addition, such an interface should provide mechanisms to check which firmware is currently installed and which configuration parameters are used.

Even if systems are equipped with an interface for applying updates, the maintenance cost can still be enormous if an administrator has to interact with each device physically and the systems are located in areas where reachability is limited. If a system is already able to communicate over a network interface, this can be leveraged to apply updates on these system - this is typically referred to as *Over the Air (OTA)*. By reusing the existing communication channels, the dedicated update interface can be omitted, which leads to smaller packaging and reduces production cost. It also decreases the maintenance cost drastically, because updates can be triggered remotely. *OTA* updates enable administrators to apply automation methods on the update process allowing to roll out new releases and fixes in a controlled fashion. As an example, updates can be done on test-devices first, followed by security-critical deployments and subordinate ones can be delayed to times when the device is not utilized. Further, a feedback channel, which provides information about the update status of a devices allows administrators to apply monitoring techniques ensuring all updates are installed and devices are in the desired state.

The remaining part of this paper is laid out as follows. Section II discusses related work. Next, in Section III we present the environment, our research is based on, while Section IV defines the requirements for the implementation of an *OTA* update mechanism in this environment. A concept for the implementation is presented in Section V and a reference implementation can be found in Section VI. Finally, a conclusion and future work can be found in Section VII.

II. Related Work

Wireless sensor and actor networks are a crucial elements of today's effort to support and implement *Industry 4.0* architectures and modern manufacturing processes. Small programmable logic controllers (PLC) and cloud computing are enabler but also drivers of these new manufacturing paradigms[1]. Thus, the networked interconnection of everyday objects, the automation of home appliances and environmental metering and monitoring based on sensor and actor networks controlled by ESP-based chipsets are subject of current research. In [2], a low-cost multipurpose wireless sensor network using *ESP8266* PLCs is introduced. The usage of *ESP8266* PLCs in combination with Raspberry PI acting as base station for the sensors is discussed in [3]. The article [4] presents a home automation solution based on a *MQTT* message queue with *ESP8266*-based sensors and actors. The control of smart bulbs with PLCs is summarized in [5]. Unfortunately, soft ware update mechanisms are not addressed in these publications. The importance of regular security updates for today's infrastructures is summarized in [6]. An approach of decentralized software updates in Contikibased IoT environments are introduced in [7]. In [8], a software update solution for devices able to execute a Java Virtual Machine (JVM) is introduced. Both solutions are not applicable for small MCU devices. In [9], a diagnoses and update system for embedded software of electronics control units in vehicles is introduced. Secure firmware updates targeted for the automotive industry is introduced in [10]. Furthermore, a secure The *Over the Air* programming capabilities of the *ESP8266* PLCs are described in [11].

III. Environment

The research presented in this paper was mainly driven by *Magrathea Laboratories e.V.* [12], the local hackerspace in Fulda, Germany, in cooperation with researchers at the department for computer science at Fulda University of Applied Sciences. Requirements were clearly defined by Magrathea Laboratories' demands to provide local and remote control over various sensors and actors in the foundations rooms to visitors and members. Such components include door sensors, power sockets, temperature sensors, projectors and screens who are all managed by a home-automation controller, which is driven by the software *home-assistant* [13]. It provides direct control over all existing components using a web-based user interface and allows to define rules and automations on how these components interact.

For the component's hardware, boards based on the *ESP8266* [14] micro-controller are used. These boards feature a small and robust design, achieve very low power consumption and integrate WiFi without requiring any extra components. It integrates a Tensilica L106 32-bit micro controller unit (MCU) with a maximum CPU performance of 160 MHz, 64 kB instruction memory and another 96 kB of main memory. According to the manufacturer, the ESP8266 is among the most integrated WiFi-capable chips in the industry. While at the beginning of this research, mostly *ESP-01s* [15] boards in combination with self-developed power supplies and use-case specific hardware components were deployed, *Sonoff* [16] wireless smart switches product series offered by *ITEAD* have been integrated quickly.

The firmware for all of the *ESP8266*-based devices in the hackerspace has been implemented using a common software platform, referred to as *ESPer*. *Sming* [17], which in turn is based on the open-source software development kit (SDK) for *ESP8266*, provides the base library for this framework. It integrates a lot of other software components and provides all kinds of functionality shared by all devices, allowing to reuse parts of the source code in multiple devices.

For communication with the controller, the *Message Queue Telemetry Transport (MQTT)* [18] protocol is used. It provides a lightweight messaging mechanism implementing the publish-subscribe pattern that allows devices to listen for commands and publish their current state to the controller and other interested parties. The controller software has out-of-the-box support for this protocol, which allows easy integration of all different device types using the same patterns.

The components all share the same configuration in regard to the network access and the controller to communicate with. The configuration is provided during build time, which eschews the need for a configuration interface and reduces the management overhead, thus minimizing security leaks.

IV. Requirements

For the implementation of an OTA update mechanism, the following requirements were defined.

1) The systems must be able to perform updates on the release of new software without manual interaction. If a new firmware version is published for a type of devices, the target devices must fetch and install the new software version automatically, and start using it subsequently if no errors have occurred during the update.

2) To ensure minimal maintenance effort, the update process should be insusceptible to errors as much as possible. Even if the installation of an update fails while reprogramming the device, the system should continue to work fully functional immediately and after reboot.

3) Firmware downloads must be possible over the same WiFi connection as used during normal operation. Fetching the firmware should be done side-by-side with operational traffic.

4) The update process must be possible over any untrusted wireless network or Internet connection. To prevent possible attackers from injecting malicious software into the embedded devices, a cryptographic signature mechanism must be implemented. New firmware only gets accepted by the device, if the cryptographic signature of the downloaded firmware image can be verified.

5) To reduce network load and aim for the maximum possible uptime of the device, the update process should only be done if a new firmware version is available. In contrast, on the release of new firmware, the roll-out to all devices should be performed as fast as possible. While checking for available updates and downloading such an update, the device should continue to work as usual.

6) For easy maintenance and monitoring, each device must provide information about the currently installed firmware version and other details relevant for the update process.

7) Devices are categorized by types. Each type runs the same software and therefore provides the same functionality. As the device type is hardly coupled to the hardware and the software interacts with it on a specific way, the update process must ensure that the correct firmware is used while reprogramming.

V. Concept for implementing *OTA* updates

To implement *OTA* updates under the given requirements, we first define a topology that integrates our build infrastructure, firmware repository, and controller with the IoT WiFi network, which the devices are connected to. For our reference implementation, we particularly chose lightweight and common software projects to allow for easy exchangeability of the individual components. The base topology, as well as the specific components used is shown in Figure 1.

The source code of the *ESPer* project is published into a *Git* [19] source code repository. From there, the continuous integration (CI) system is responsible for automatically building and publishing the firmware image files, as soon as updated source code is available. It is also in charge of assembling and publishing meta-information consisting of version number and cryptographic signature required for the update process. The CI systems is described in detail in the following section. Updates to the devices firmware are either triggered actively (i.e., manual or by the CI) or on a regular schedule by the devices themselves. This process is described in Section V-B.

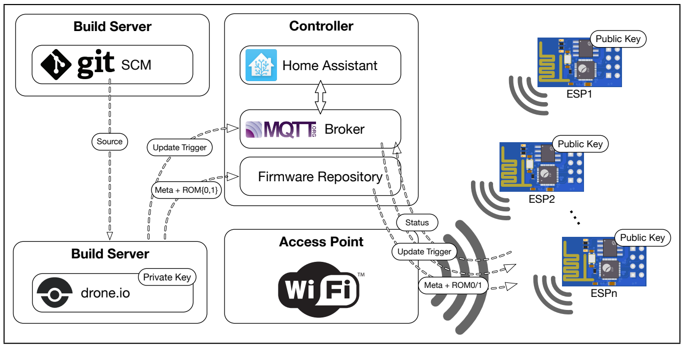


Figure 1. The base network topology.

For monitoring and maintenance purposes, each device publishes a set of information to a well-known *MQTT* topic after connecting to the network. Beside data like device type, chip and flash ID, the published data includes details about the bootloader, SDK and firmware version as well as relevant details from the bootloader configuration, like the currently booted ROM slot and the default ROM slot to boot from. This allows administrators to find devices with outdated bootloaders and helps to find missing or failed updates.

A. Common framework and build infrastructure

The framework includes a build system, which allows to configure basic parameters for all devices, including, but not limited to, the WiFi access parameters, the *MQTT* connection settings and the updater URLs. Each device requires to have the `UPDATE_URL` option set to make the update work. Skipping the option results in the exclusion of the code for update management during the build. By sharing the same code, all devices ensure to have a common behavior when it comes to reporting the device status or interacting with the home-automation controller. This eases configuration and allows to collect information about all devices at a central location.

As development on the devices usually happens in cycles, some of the projects would miss updates of the framework and therefor would not benefit from newly added features or fixed problems. Regularly updating the framework version and rebuilding the firmware would often result in an easy gain of these benefits, but requires manual interaction. Further, problems could arise if the application programming interface (API) of the framework changes. In this situation, the device firmware must be updated to use the changed API, which can be an unpleasant and complex task that leads to higher latency for firmware updates. To prevent these problems, the firmware of all devices in the hackerspace is integrated together with the framework into a larger project. By doing so, any device specific code is always linked to the latest version of the framework. The according device type is provided as a string through a global constant at compile time and it must never be changed during operation. Device specific code is organized in a sub-folder for each device type. To build the software, a *Makefile* [20] is used, which provides a simple way for reproducible builds. Whenever a new build is started, the build system scans for all device specific folders and calls the build process for each of them. After the build of the firmware has finished, the build system also creates a file for each device type, containing the build version and cryptographic signatures of the corresponding firmware images. To avoid interferences between different build environments, and to roll out new versions as quickly as possible, the code has been integrated into a CI system, which is also responsible for publishing the resulting firmware images to the firmware server queried during updates, and for notifying the devices to check for an update.

B. Device setup and flash layout

Microcontroller boards based on the *ESP8266* MCU are mostly following the same layout: the MCU is attached to a flash chip, which contains the bootloader, firmware and other application data. The memory mapping mechanism of the MCU allows only a single page of 1 MB of flash to be mapped at the same time [21] and the selected range must be aligned to 1 MB blocks.

As the firmware image to download and install possibly exceeds the size of free memory heap space, the received data must be written to flash directly. In contrast, executing the code from the memory mapped flash while writing the same area with the downloaded update leads to unexpected behavior, as the executed code changes immediately to the updated one. To avoid this, the flash is split into half to contain two firmware ROM slots with different versions, one being executed and one which is being downloaded (see Figure 2). In addition to the two firmware ROM slots, the flash provides room for the bootloader and its configuration. For alignment and easy debugging, the second block is shifted by the same amount of bytes as the first block. The gap of 8192 bytes is available to applications to store data, which can persist over application updates.

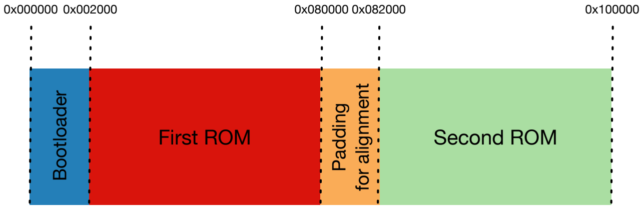


Figure 2. The flash layout used for two ROM slots.

This standby ROM slot also acts as a safety mechanism if the download fails or is interrupted as the previous version stays intact and can still be used (refer to requirement IV-2).

C. Cryptographically securing the firmware update

To ensure only valid firmware is running on the devices, a cryptographic signature of the firmware images is calculated and checked as part of the update process. For calculating and verifying the signatures of a firmware image, the *SHA-256* hashing algorithm [22] and an elliptic curve cipher based on *Curve25519* [23] are used, which are both considered modern and secure methods for software signing (see [24], [25]). The cryptographic signature for each of the two firmware images is created by the continuous integration system during build time and is provided as meta-information along with the firmware images. Therefore, the CI system must be equipped with the private key used to create the signatures. In contrast, to be able to verify the cryptographic signature the micro controller only needs to know the according public key. For the same reason as stated in Section V-B, the signature of the new firmware image can not be verified before it is written to flash.

Therefore, the calculation of the *SHA-256* checksum required for the signature check is done while the update is downloaded and written to flash. After the download has succeeded, the checksum is verified against the signature and the bootloader gets reconfigured iff the signature is validated successfully. Otherwise, the bootloader will not be reconfigured and the system will not start the invalid firmware.

VI. Implementation

Implementing *OTA* updates under the given requirements involves multiple components, which interact closely. The continuous integration system is in charge of building the firmware from source, calculating cryptographic signatures, and publishing the built firmware images. The deployment infrastructure provides resources for downloading the firmware images and triggering the update on all devices. Finally, the implementation of the update mechanism, as a part of the firmware running on the embedded device, is responsible for downloading and installing the updates.

A. *Build infrastructure and automatic deployment*

The CI system, which is based on *drone* [26] allows to execute commands, whenever a new version is published into the projects *Git* repository. A corresponding *drone* configuration called `.drone.yml` exists beside the source code (Figure 3). Within this configuration file, settings relevant to the build process are provided to the build environment. First, the `CONFIG=maglab` option lets the build system use an additional configuration file (`Configurion.mk.maglab`), which is stored inside the framework repository and provides environment specific information, such as the WiFi SSID. To keep secrets like the WiFi password and the private key unexposed, it is not written down in the configuration file. Instead, to include secrets into a build process while allowing to keep the configuration public, *drone* allows to encrypt these with a repository specific key. Using this method, the secrets are stored as `.drone.sec` file inside the repository from where they are injected into the build environment. Also noticeable in Figure 3 is the firmware version, which is configured to be the first 8 letters of the *Git* commit hash uniquely identifying a version of the source code. For deployment, only the master branch is considered. After a successful build, all distribution files (the firmware image and meta-information files) of all device types are copied to the repository server, from where they are served by a *HTTP 1.1* [27] server. The configuration file (`Configurion.mk.maglab`) references exactly this repository server as the source for updates.

```
build:
  image: maglab/sming
  environment:
    - CONFIG=maglab
    - WIFI_PWD=$$WIFI_PWD
    - VERSION=$${COMMIT:0:8}
  commands:
    - make clean && make
```

Figure 3. The *drone* configuration for the *ESPer* project.

Support for multiple devices of different type is implemented in both, the *ESPer* framework itself and the build system. The framework keeps control over the application lifecycle. It ensures that device unspecific code is executed at the right time and provides an API for device specific functionality. For this, a simple interface is specified by the framework, which must be implemented by each device. A single function `Device* getDevice()` must be defined exactly once in each device specific folder. To implement this interface, a static instance of `Device` is created and returned. Each `Device` is populated with device specific `Feature` instances. While the `Feature`-API leverages common run time polymorphism to share functionality between features, the initial `Device` creation uses compile time polymorphism, which reduces the need for memory management and increases performance by avoiding virtual function tables. Figure 4 shows the complete device specific code used for a simple power socket, which is mainly confined to the device type and its capabilities (e.g., the GPIO pin numbers to use).

```
constexpr const char NAME[] = "socket";
constexpr const uint16_t GPIO = 12; // General purpose I/O

Device device;
OnOffFeature<NAME, 12, false, 1> socket(&device);

Device* getDevice() { return &device; }
```

Figure 4. Device specific code for a socket driver.

The actual compilation of the source code is mainly controlled using two *Makefiles*. The first one is a helper *Makefile* built to accept a parameter for device type identifiers called `DEVICE`, and to create its whole output inside a subdirectory specific to the device type. In addition, the primary *Makefile* scans a project subdirectory and uses each directory in there as a container for device specific code. For each of these directories, the helper *Makefile* is called and the subdirectories name is used as the value of the `DEVICE` parameter. By splitting the build and recompiling the framework each time before intermixing it with the device specific code, the device type identifier can be used inside the shared framework code. While building a devices firmware, the meta-information file used during updates is also created and stored beside the firmware image. For development, each device can be build separately by using the device type identifier as *Makefile* target. In addition, the suffix `/flash` can be used to flash a specific firmware to the device.

While building the firmware images for a device, the build environment provides some constants, which are baked into the resulting firmware image. Beside the environmental configuration like the WiFi credentials, *MQTT* topics and other configurable tweaks, the current device and version identifiers are provided as compile time constants. In addition, the public key used to verify firmware signatures during updates is derived from the private key and provided as a object file, which is linked into each firmware image (Figure 5). This allows to use all the information inside the code without any overhead while being configurable during build time.

As the *ESP-01s* is only equipped with 1 MB of flash, this means that the whole memory is mapped to a contiguous address space (refer to Section V-B). Therefore, the second ROM slot can not be re-mapped to have the same start address as the first ROM slot. While the firmware is executed without any dynamic linking mechanism and the chip does not support position independent code, the addresses used in the ROM slots are dependent to the offset at which the firmware is stored. This arises the need for building two firmware images, one for each target location. To do so, a linker script for each of the two ROM slots was created, which is used to create two variations of the same firmware, only differing in ROM placement. The two resulting firmware image files are both provided for download via *HTTP 1.1* - which one to download depends on the target ROM slot and is selected by the device during the update process. Figure 6 shows the only difference between the two linker scripts, where `${SLOT}` is replaced with the slot number according to the current build.

```
update_key_pub.bin:
    echo "$(UPDATE_KEY)" | ecdsakeygen -p | xxd -r -p > "$@"

update_key_pub.o: update_key_pub.bin
    $(OBJCOPY) -I binary $< -B xtensa -O elf32-xtensa-le $@
```

Figure 5. Creating the linker object containing the public key.

The build process will create the two firmware images, one for each ROM slot, and the meta-information file. To create the meta-information file, the current version identifier is written to the `.version` file. After the build, the signatures for both firmware images are created and attached to the file. Due to modern compilers doing link time optimization, the resulting firmware images include only code needed according the actual configuration.

```
irom0_0_seg :
  org = ( 0x40200000          // The memory mapping address
        + 0x2010              // Bootloader code and config
        + 1M / 2 * ${SLOT} ), // Offset for the ROM slot
  len = ( 1M / 2 - 0x2010  )  // Half ROM size excl. offset
```

Figure 6. Linker script to build firmware for two ROM slots.

B. The update mechanism

The update mechanism is split into four main phases: checking for updates, reprogramming the device, calculating and verifying the cryptographic signature of the updated firmware, and - assuming that the update was successful - reconfiguring the boot process to use the new firmware.

1) Checking for updates: In order to inform the IoT devices of the availability of a new firmware version, the update server provides a file for each device type containing meta-information about the latest available firmware version. The meta-information file has a simple line oriented ASCII format, which is easy to generate and efficient to parse within the limited constraints of the embedded device. It consists of the version identifier and the cryptographic signatures of both of the firmware binaries. The version identifier can be an arbitrary string as the content is not interpreted semantically but only compared to the version identifier used during build time. The other two lines in the meta-information file provide the hexadecimal representation of the cryptographic signatures, one line for each firmware binary file. These meta-information files are provided by the update server using *HTTP 1.1* [27] under the following path pattern: `${DEVICE}.version` (whereas `${DEVICE}` gets replaced by the device type name). Each device queries the update server regularly for the currently available firmware version. It uses the `UPDATER_URL` option to identify the update server. After the meta-information file has been downloaded successfully, the version identifier is extracted and compared to the version identifier of the running firmware. If the version identifiers differ, the update process is initialized. In cases where the download fails, the update server or network connection is not available, or any other error occurres, another attempt will be made automatically at the next regular interval. In addition to the interval, a special *MQTT* topic shared by all devices is subscribed on device startup: `${MQTT_REALM}/update`. Every time a message is received on this topic, a fetch attempt for the meta-information file is triggered and the process restarts. This allows faster roll-outs of updates and finer control for manual maintenance.

2) Reprogramming the device: The firmware files provided on the update server are the exact same ones as used to initially flash the chip for the according version. Using the same files for flashing and updating allows better debugging by eliminating errors related to the update process itself and eases development and initial installation. Figure 7 shows the algorithm used to determine the download address and reconfigure the bootloader. The update server provides these files in the exact same way as it provides the meta-information files, but the path pattern differs: the suffixes `.rom{0,1}` are used to provide the firmware image files for the first and second slot respectively. For installing a firmware update, the new firmware image file is downloaded using an *HTTP 1.1* `GET` request.

```
#define URL_ROM(slot) (( URL "/" DEVICE ".rom" slot ))

// Select rom slot to flash
const auto& bootconf = rboot_get_config();
if (bootconf.current_rom == 0) {
  updater.addItem(bootconf.roms[1], URL_ROM("1"));
  updater.switchToRom(1);
} else {
  updater.addItem(bootconf.roms[0], URL_ROM("0"));
  updater.switchToRom(0);
}
```

Figure 7. Configuring the updater to download the right firmware image and update the booloader accordingly.

3) Verifying the cryptographic signature: While the image is being downloaded, each chunk received in the download stream is used to update the *SHA256* hash before it is written to the flash. When the write has been finished, the next chunk is received and the process continues until all chunks have been processed. After downloading the new firmware image has been finished successfully, the calculated hash is checked against the signature of the according firmware image. Therefore, the cryptographically signed hash, which was provided in the meta-information file triggering the update, is verified against the *Curve25519* public key stored as a constant in the running firmware. Only if the checksum matches the provided signature, the firmware is considered valid and the process is continued.

4) Reconfiguring the boot process: For the bootloader, *rBoot*[28] has been choosen as it is integrated within the *Sming* framework and allows to boot to multiple ROM slots. For configuration, an *rBoot* specific structure is placed in the flash at a well-known location directly after the space reserved for the bootloader code. This structure contains, among other things, the target offsets for all known ROM slots and the number of the ROM slot to boot on next startup. To switch to the updated ROM slot after successful installation, the number ROM slot to boot on startup is changed in the configuration section and the device is restarted.

VII. Conclusion

In this article, we have presented a concept for building and publishing cryptographically secure *Over The Air* updates for embedded devices based on ESP8266 microcontrollers. A proof of concept implementation has been developed, which is now an essential part of the home-automation development and deployment in the *Magrathea Laboratories e.V.* hackerspace. All of the devices running the OTA-enabled firmware have undergone multiple major updates without any problems. This includes a major network configuration change and an important stability fix for the network communication stack. All devices applied the update successfully and started to work without any manual interaction required afterwards.

While the devices from various manufacturers in the hackerspace are all delivered with a pre-installed firmware, which is thought to be ready for smart home application, none of them has been provided with updates by the manufacturer so far. It is not visible to the users if the current firmware of these devices is at the latest version nor which versions are installed or how to update them.

The update infrastructure has been the crucial point for most of our members towards the framework. Enabling the developers to do updates in combination with the shared configuration and behavior provided by the framework resulted in a massive speedup when it comes to project deployment. Before that, the cost for applying changes after deployment was estimated so high, that most projects tend to delay deployment until all required and wanted features were implemented. Now, as the devices are deployed as soon as the hardware is considered stable, these devices start to provide functionality early and therefore the developers can get better feedback on the provided functionality.

The project will be continued to extend the functionality and security with features already being in development. The latest development includes further security enhancements by implementing checksum verification during startup where the hash of the firmware image is checked on each boot by the bootloader to detect tempering and defects. It also considers including the device identifier into the signature to prevent confounding of images between different device types. Last, the standby ROM slot will be updated right after each successful update to be more failsafe.

In addition, the information provided by the device about the firmware status will be enhanced to allow better control and reduce maintenance effort even more. A web interface to review the published information is currently in development.

References

[1] “Industry 4.0: A Cost and Energy efficient Micro PLC for Smart Manufacturing,” Indian Journal of Science and Technology, vol. 9, no. 44, Nov. 2016.

[2] “Design of a low cost multipurpose wireless sensor network,” in 2015 IEEE International Workshop on Measurements and Networking (M&N). IEEE, 2015, pp. 1–6.

[3] “ESP8266 based implementation of wireless sensor network with Linux based web-server,” in 2016 Symposium on Colossal Data Analysis and Networking (CDAN). IEEE, 2016, pp. 1–5.

[4] “MQTT based home automation system using ESP8266,” in 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, 2016, pp. 1–5.

[5] “An IOT by information retrieval approach: Smart lights controlled using WiFi,” in 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence). IEEE, 2016, pp. 708–712.

[6] A. R. Beresford, “Whack-A-Mole Security: Incentivising the Production, Delivery and Installation of Security Updates,” in IMPS@ ESSoS, 2016, pp. 9–10.

[7] P. Ruckebusch, E. De Poorter, C. Fortuna, and I. Moerman, “GITAR - Generic extension for Internet-of-Things Architectures enabling dynamic updates of network and application modules.” Ad Hoc Networks, 2016.

[8] “Decentralized coordination of dynamic software updates in the Internet of Things,” in 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT). IEEE, 2016, pp. 171–176.

[9] K. Mansour, W. Farag, and M. ElHelw, “AiroDiag: A sophisticated tool that diagnoses and updates vehicles software over air,” in 2012 IEEE International Electric Vehicle Conference (IEVC), year = 2012, pages = 1–7, publisher = IEEE.

[10] D. K. Nilsson and U. E. Larson, “Secure Firmware Updates over the Air in Intelligent Vehicles,” in ICC 2008 - 2008 IEEE International Conference on Communications Workshops. IEEE, 2008, pp. 380–384.

[11] S. Gore, S. Kadam, S. Mallayanmath, and S. Jadhav, “Review on Programming ESP8266 with Over the Air Programming Capability,” International Journal of Engineering Science, vol. 3951, 2016.

[12] Magrathea Laboratories e.V., “Magrathea Laboratories - Creating new Worlds,” URL: https://maglab.space/, [accessed: 2017.05.22].

[13] Home Assistant, “Awaken your home,” http://home-assistant.io/, [accessed: 2017.05.22].

[14] ESPRESSIF, “ESP8266 Overview,” URL: http://www.espressif.com/en/products/hardware/esp8266ex/overview, [accessed: 2017.05.22].

[15] SparkFun, “WiFi Module - ESP8266,” URL: https://www.sparkfun.com/products/13678, [accessed: 2017.05.22].

[16] ITEAD, “Sonoff Smart-home,” URL: https://www.itead.cc/smart-home.html, [accessed: 2017.05.22].

[17] Sming, “Sming - Open Source framework for high efficiency native ESP8266 development,” URL: http://sminghub.github.io/Sming/about/, [accessed: 2017.05.22].

[18] OASIS Standard Incorporating, “MQTT Version 3.1.1 Plus Errata 01,” URL: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html, [accessed: 2017.05.22].

[19] git, “git - a free and open source distributed version control system,” URL: https://git-scm.com, [accessed: 2017.05.22].

[20] The IEEE and The Open Group, “The Open Group Base Specifications Issue 6 - make - maintain, update, and regenerate groups of programs,” URL: http://pubs.opengroup.org/onlinepubs/009695399/utilities/make.html, [accessed: 2017.05.22].

[21] E. Community, “ESP8266 Memory Map,” URL: http://www.esp8266.com/wiki/doku.php?id=esp8266_memory_map, [accessed: 2017.05.22].

[22] D. Eastlake and T. Hansen, “US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF),” Internet Requests for Comments, RFC Editor, RFC 6234, May 2011, http://www.rfc-editor.org/rfc/rfc6234.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6234.txt

[23] D. J. Bernstein, “Curve25519: new Diffie-Hellman speed records,” in International Workshop on Public Key Cryptography. Springer, 2006, pp. 207–228.

[24] E. Barker and Q. Dang, “NIST Special Publication 800–57 Part 1, Revision 4,” 2016.

[25] F. O. for Information Security, “Cryptographic Mechanisms: Recommendations and Key Lengths,” Online, Federal Office for Information Security, BSI Technical Guideline BSI TR-02102-1, February 2017, URL: https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf, [accessed: 2017.05.22].

[26] Drone, “Drone is a Continuous Delivery platform built on Docker, written in Go,” URL: https://github.com/drone/drone, [accessed: 2017.05.22].

[27] The Internet Society, “Hypertext Transfer Protocol – HTTP/1.1,” URL: https://www.w3.org/Protocols/rfc2616/rfc2616.html, [accessed: 2017.05.22].

[28] R. A. Burton, “An open source bootloader for the ESP8266,” URL: https://github.com/raburton/rboot, [accessed: 2017.05.22].

Multi-Dimensional Detection in Future Hyper-Scale Datacenters

Santiago Echeverri-Chacón

Department of Photonics Engineering
Technical University of Denmark
2800 Kgs. Lyngby, Denmark

Mellanox Technologies Denmark ApS
4000 Roskilde, Denmark
e-mail: santiagoe@mellanox.com

***Abstract*— Extraordinary demand of internet services is challenging the growth capacity of datacenters and their networks. In particular, interconnect suppliers will need to explore radical solutions in order to keep up with bandwidth expectations from service providers. In this paper, we discuss key paths towards multi-dimensional detection in interconnect channels that will be relevant for hyper-scale datacenters. The emphasis is on highlighting challenges and opportunities of receiver architectures that have recently been pitched to replace intensity modulation and direct detection in the datacenter interconnect scenario.**

Keywords- Fiber optics; Optical communications; Datacenter interconnection.

I. Introduction

There is a strong demand for delivering interconnect solutions for next-generation hyper-scale datacentre interconnects that support 400 Gb/s and beyond [1]. Technologies that support higher symbol rates per wavelength channel will be critical to connect bigger campuses and regional clusters demanding more efficiency in fibre deployment.

A typical datacentre network is shown in Figure 1. Data Centre Interconnects (DCI), provide connectivity for networking, storage and compute resources inside (intra-DC) or between (inter-DC) datacentres and are based on optical or copper channels. Passive copper cables also known as Direct-Attach-Copper (DAC) are still the most effective solution for links of < 5 m which are used to connect servers to the Top-of-Rack (ToR) switch. Connections from the ToR switch to other layers of the network, as well as all other DCIs are based on optical signalling. Both multimode (MMF) [2] and single mode (SMF) [3] optical fibres are used. Transceivers for use with MMFs have arrays of Vertical-Cavity Surface-Emitting Lasers (VCSELs) and are commonly used for driving channels of lengths within 5 - 100 m linking switches within a room. The vertical design of VCSELs allows them to be produced, tested and packaged at a low cost, while their circular cavity permits low insertion loss when coupling to the core of MMFs. On the other hand, SMF solutions require edge emitting lasers and optical assemblies tailored to couple light to a fibre with a much smaller core. Excitation of a single mode enables communications over longer fibre spans and opens the door to Wavelength Division Multiplexing (WDM) as a technique to parallelize data transmission. Within the datacentre, transceivers for links between a hundred meters and 2 km are increasingly implemented by photonic integrated circuits coupled to arrays of Parallel Single Mode Fibre (PSM4).

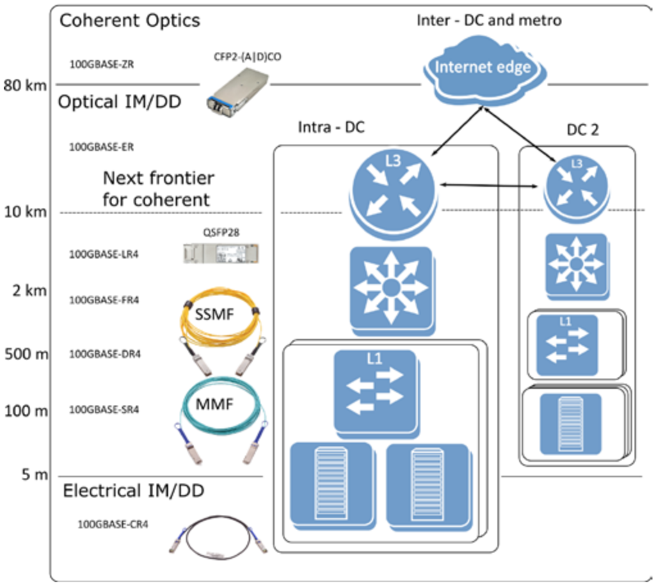


Figure 1. Datacentre architecture and interconnect solutions for inter- and intra-data centre links.

Longer reach solutions connecting campuses (2-80 km) use externally modulated lasers and optical elements for WDM, which are hosted inside hermetically-sealed transmitter and receiver optical sub-assemblies.

Optical interconnects below 80 km use Intensity Modulation and Direct Detection (IM/DD) of optical signals and are so far dominated by the Non-Return to Zero (NRZ) modulation format. Nevertheless, the industry is rapidly migrating from NRZ to Multilevel Pulse-Amplitude Modulation (M-PAM), which has higher Spectral Efficiency (SE) [4][5]. The format with 4 levels, PAM4, is the first to gain broad support and will most likely become and remain dominant in the short term. When looking further ahead in time, overcoming channel impairments on links above 10 km will become difficult without signal processing techniques that are comparable in complexity to more efficient alternatives.

Other advanced modulation formats achieve better sensitivity and SE than PAM4 by taking advantage of higher dimensionality and can become increasingly attractive to increase bitrates in the future, without adding extra channels in the form of wavelengths or fibres. However, these advantages require receiver architectures that are significantly more complex than in DD.

This paper reviews some of the paths towards implementing multi-dimensional receiver solutions in DCI. This vision is inspired by the current transition between IM/DD and coherent detection in the metro scenario enabled by novel coherent pluggable transceivers, such as the 100 Gb/s (C) Form-Factor Pluggable version 2 (CFP2) [6].

The manuscript is organized in the following manner: In Section 2 the need for architectures that overcome the limitations of IM/DD in our scenario is stated, and a brief overview of the selection of candidate approaches is presented. Section 3 provides a review of each receiver architecture and highlights implementation challenges. Finally, the work is concluded in section 5.

II. Technologies for Datacenter Interconnects

The relatively low cost of adding fibres, and the advent of WDM has kept IM/DD solutions dominant in the DCI scenario; even though, bitrates per wavelength have remained unchanged at 25 Gb/s per lambda solutions. A possible explanation for the tendency towards multiplexing is the slow progress in the development of production-ready optical building blocks with high bandwidth, low noise and sufficient linearity. It is hinted that the transition to multi-level signalling in the form of PAM4 at high symbol rates (50 GBd) will likely require Digital Signal Processing (DSP) to enable increasingly elaborate equalization techniques [7]. DSP will in turn bring power and heat dissipation issues that need to be resolved to fit the stringent specifications of next generation transceiver modules. In addition, there are limits to: the number of fibres, the fabrication tolerances of WDM, and the power needed to support multiple channels. These limits on symbol rate and spectrum will encourage bitrate increase by means of modulation formats that use other orthogonal dimensions for coding [8].

Well-known dimensions for coding in SMF channels include the phase and linear State of Polarization (SoP) of the optical field. Commercial coherent receivers detect the phase and amplitude of each SoP, and will be regarded here as able to decode in 4 orthogonal dimensions or “4D”. There is also interest in alternative receiver architectures capable to recover signals in 3D and 2D that may be a better fit for the specific conditions of DCIs.

A. *Overview of architectures*

Figure 2 presents a selection of multi-dimensional receiver architectures that can be pitched to bring multidimensional signalling in the DCI scenario. They are ordered by decreasing level of trade-off between complexity and spectral efficiency. Additionally, selected elements of each architecture have been coloured according to the level of uncertainty or risk needed to develop a mature solution. Red coloured blocks indicate a significant challenge, where either elements are still too complex, or unfit for the DCI requirements. Yellow coloured blocks indicate that there is a mature assessment of the limitations and evidence of strategies to overcome them.

We begin with a light-DSP version of the mainstream polarization division multiplexing intra-dyne coherent approach shown in Figure 2(a). The objective is to further tailor traditionally power-hungry optics and algorithms for the specific conditions of short reach communications.

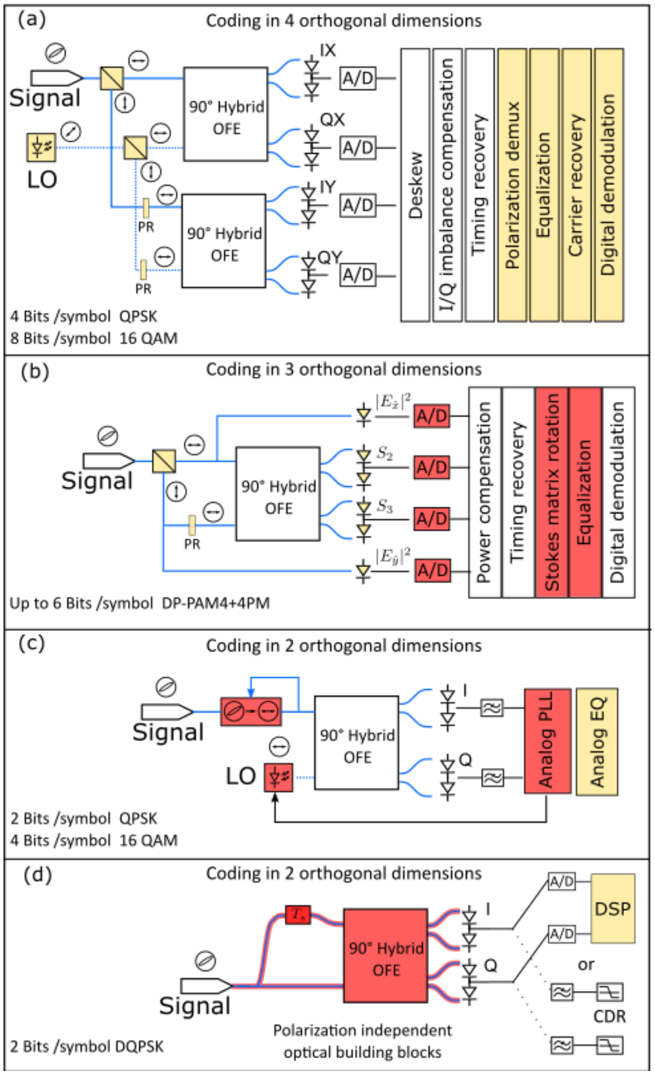


Figure 2. A selection of ordered multi-dimensional receiver architectures for future datacenter interconnects by decreasing complexity: (a) Intra-dyne digital coherent receiver. (b) Stokes vector receiver. (c) Homodyne coherent receiver and (d) self-homodyne differential receiver.

Then, Figure 2(b) illustrates the basic architecture of Stokes Vector Receivers (SVR) [8] which demodulate signals encoded in the power (or Stokes) space. Signals modulated in the Stokes space are independent of the absolute phase of the carrier, but dependent on the relative phase and amplitude ratio between the vertical and horizontal polarization components [9][10].

An SVR as shown here can demodulate signals encoded in up to three orthogonal dimensions, and achieve spectral efficiencies of up to 6 bits per symbol per wavelength.

One can also consider technological developments that revive single-polarization architectures from early coherent demonstrations. Figure 2(c) represents homodyne coherent receivers where analog electronics are used to track variations in the phase of the carrier. Here, we have stressed the importance of including active polarization tracking.

Finally, Figure 2(d) illustrates a concept of single polarization coherent receiver that uses self-homodyne detection and differential coding to achieve polarization-independent multi-dimensional signalling on the phase and amplitude.

III. IMPLEMENTATION CHALLENGES OF ARCHITECTURES

The objective of this section is to go through the architectures that were briefly mentioned above and highlight some of the specific challenges and advantages of each.

Given that photodiodes can detect only optical power, interferometric techniques are used to obtain the phase of a signal by analysing the effects on the intensity at the outputs of a phase diversity measurement where the modulated signal is mixed with a reference signal [11]. All Optical Front Ends (OFE) in Figure 2 share the building block known as a 90° hybrid used for optical mixing, but differ in the way in which information is extracted from measurements. Additionally, they all use some form of homodyne detection, meaning that signals at the input of 90° hybrids should strive to have the same frequency. Finally, a last critical condition for mixing is the control of the SoP. This is because most integrated optics elements are polarization dependent and received signals have an unknown SoP due to random birefringence in the fibre. Different approaches to deal with polarization control are shown before the 90° hybrids and will be mentioned in what follows.

A. *Intra-dyne digital coherent receivers*

The OFE in commercial coherent receiver architectures has remained unchanged in the form of the intra-dyne digital coherent receiver in Figure 2(a). It is referred to as intra-dyne because small variations in frequency and phase between the signal carrier and Local Oscillator (LO) are corrected using DSP. Powerful DSP Application Specific Integrated Circuits (ASICs) enabled not only algorithms for Carrier Recovery (CR) and channel impairment compensation, but also techniques for digitally tracking the state of polarization. A trade-off of coherent Polarization Division Multiplexing (PDM) is the doubling in complexity of the OFE. To extract phase information from independent polarizations, polarization beam splitters and rotators project the fields (signal and LO) into the known orthogonal basis of the 90° hybrids. The considerable number of elements, combined with the need of a LO has historically justified scepticism in short reach intra-dyne coherent solutions.

The top-down approach for developing receivers like the one in Figure 2(a) for gradually shorter links is the one most favoured by the industry. Challenges are well known and there is consistent progress in reducing the power consumption and size of optical, thermal and electronic components. The tradition inherited from the long haul encourages multi-level modulation formats like 16 Quadrature Amplitude Modulation (QAM) or even 64-QAM for short reach to better use the channel capacity given a larger Signal-to-Noise Ratio (SNR) [12]. But this approach assumes keeping the DSP complexity and could be challenged by bottom-up implementations that focus on symbol rate and simplicity instead of channel capacity.

Photonic integration of optical components is critical for reducing the costs, size and power consumption of the optical frontend [13]. Silicon photonics in particular, promises receiver solutions with low loss, no need for hermetic sealing, and can profit from the economics of scale and maturity of established CMOS foundries. In addition, advances in the supporting technologies of components, such as lasers, phase shifters and detectors, could further reduce the complexity of equalization and CR routines in the digital domain.

The complexity of ASICs is dominated by algorithms for compensation of channel impairments, such as Chromatic Dispersion (CD) and Polarization Mode Dispersion (PMD) that decrease in magnitude with shorter fibre lengths typical of the DCI scenario. It is conceivable that big reductions in complexity could be achieved by tailoring ASICs for the next frontier of coherent pluggable, on the condition of sufficient demand of spectral efficiency in the datacentre. Moreover, ASICs have been following Moore’s law enabling ×0.7 power savings per process node every two years [12]. A tailored solution for short reach could also loosen vertical resolution requirements on elements for Analog-to-Digital Conversion (ADC) and support implementations with simple constellations at a high symbol rate.

B. *Stokes vector receivers*

Modern Stokes vector receivers are developments of early attempts to implement PDM of IM/DD signals and are generally categorized as a form of direct detection. Direct detection approaches for PDM, including early versions of SVR, lost the race against coherent detection in the long haul when the opportunity to do linear digital processing on the later demonstrated its superiority for channel impairment compensation. However, there has been a revived interest in SVR [8]–[10] for optical channels that are too long for PAM4 and too short to justify intra-dyne coherent detection. These efforts have now resulted in working implementations of 2D formats like Dual Polarization PAM4 (DP-PAM4), and 3D formats, such as DP-PAM4 with inter-polarization Phase Modulation (DP-PAM4+PM). Thus, one advantage of SVRs is the possibility of detecting formats with SE between 2-6 bits/symbol/wavelength and optimize complexity for a given channel length.

Two of the DSP blocks in Figure 2(b) are highlighted in red because algorithms for de-multiplexing and equalization after the OFE have not converged to a standardized form. They have also not been optimized and synthesized for use in ASICS as is the case for the others. It is not yet clear what is the expected power consumption of a SVR ASIC and how it compares to state-of-the-art coherent DSP. The ASIC could be less complex than an intra-dyne equivalent if implemented for short reach communications where CD and PMD are not significant. Additionally, in demonstrations of multi-dimensional detection using SVR, high performance ADC and DAC blocks have been necessary to showcase high bitrate operation. This suggests that there is a requirement for extra sensitivity given the square root power envelope conditions of direct detection.

Photonic integration is already playing a role in realizing specialized building SVR blocks for next generation multi-dimensional enabled short-reach links [14][15]. It is worth noting that compared to the architecture in Figure 2(a) the SVR has only one 90° hybrid, half the number of polarization beam splitters and rotators, no LO and only 6 PDs of which two are single ended.

C. Homodyne coherent receivers

In homodyne coherent receivers, the LO must track the frequency and SoP of the incoming signal. Analog receivers electronically detect variations in frequency and use electrical feedback signals to lock the LO laser to the carrier frequency. Recent implementations of analog Phase Lock Loops (PLL) [16]–[18] present homodyne receivers working at still modest speeds of 40 Gbit/s. Similarly, integrated polarization controllers are slowly becoming a reality [19]–[21]. Additionally, analog equalization has also been demonstrated recently at low symbol rates [22]. Sufficient innovation in integration and co-packaging could revive single polarization homodyne coherent detection and take full advantage of the cost and power efficiency of analog electronics for the short reach scenario.

Variations on single polarization coherent detection using a LO can also be found in coherent access networks [23], which as the DCI scenario, are also constrained by cost and low power requirements on the LO and DSP. Noteworthy, are heterodyne solutions using 3x3 optical couplers [24]. Other novel ideas based on homodyne detection and inspired in access networks, use carrier delivery for remote modulation in a bi-directional link [25].

D. Delay-line based differential coherent receivers

Delay-line based differential coherent receivers [26], like the one depicted in Figure 2(d), are one important group of receiver architectures known as Self-Coherent (SC). A common trait of self-coherence is the absence of a LO, which allows avoiding carrier recovery schemes. An excellent comparison of SC receivers can be found in [27].

In delay-line receivers, the difference between the phases of consecutive symbols is detected instead of the absolute phase of the field. Detection is achieved by mixing the signal with a delayed copy of itself. Provided that the OFE has a polarization controller or that it can be made of polarization independent optical elements, the SC receiver can recover phase and amplitude modulated signals on a single polarization. Additionally, even though most demonstrations use DSP, analog signal processing, even if not linear, is possible [28] without the loop bandwidth constraints of a PLL or the need of polarization de-multiplexing. The apparent simplicity of this approach hides significant challenges in the design of an OFE that guarantees precise delays in both polarizations at a given wavelength.

Implementations of photonic integrated SC differential receivers have been demonstrated using Mach-Zehnder delay-line interferometers [29][30], and recently ring resonators [31]. Free-space optics based self-homodyne coherent receivers [32] solve the problems of polarization independence and tuning flexibility, but require bulky and expensive building blocks.

In other manifestations of self-coherent receivers, mixing happens at the transmitter or is done with a reference that travels alongside the signal. When mixed at the transmitter, single polarization DD approaches using only one photodiode can be implemented at the cost of higher DAC/ADC bandwidth, large processing complexity, and a high Carrier-to-Signal Power Ratio (CSPR) [33]. A similar trade-off is present when sending the carrier on an orthogonal polarization [34]–[36].

E. Note on transmitter complexity

In this section, we chose to analyse the technological challenges of receiver architectures, leaving aside the topic of multi-dimensional modulators. This is because a basic IQ modulator in combination with a polarization beam combiner is a sufficiently general system for coding in any combination of the 4 orthogonal dimensions mentioned above. Differences between modulation formats come from the driving signals sent to individual phase shifters placed in the Mach-Zehnder modulators and waveguides that make up a traditional IQ modulator.

F. Common challenges

As hinted above, common challenges are photonic integration and achieving packaging solutions where driving electronics and DSP ASICs are placed closer to the optics.

On the photonic integration aspect, the selection of substrate platform is critical. There are trade-offs to be considered when deciding to implement architectures in the III-V material ecosystem or in silicon photonics. Hence, many commercial solutions tend to favour heterogeneous integration, especially in the transmitter side. Above all, the platform on which multi-dimensional receivers are built should support polarization handling elements, such as polarization beam splitters and polarization rotators. Or alternatively, guarantee polarization independence. Not all foundries have polarization handling building blocks and some designs require extra processing steps, such as wet etching, that can increase the complexity of fabrication and reduce yield.

Regarding packaging, a significant milestone was achieved recently when Ball-Grid-Arrays (BGA) packaged coherent modules were demonstrated [37]. Finally, hybridization of III-V active components on a Silicon substrate is an ambitious objective that could significantly reduce power and footprint constraints [38].

IV. CONCLUSIONS

Multi-dimensional modulation formats are starting to be proposed to achieve better SE in the DCI scenario. We have highlighted four receiver architectures as an attempt to review trends that are relevant for solving requirements of next generation DCI links. The discussion on the specific challenges that are faced on each case will be the foundation of future decisions on a path to follow. Even though it looks like the industry is focusing on gradually reducing the complexity of traditional coherent transceivers, we have observed and presented examples of architectures such as SVR or variations of SC receivers that have the potential to move the frontier of multi-dimensional modulation formats to shorter reach applications with reduced complexity.

Acknowledgment

S. Echeverri would like to express his gratitude to Innovation Fund Denmark who partly funded this work under file No. 5189-00006B. The author also wants to thank D. Feng, and J. Levy from Mellanox Technologies (Monterey Park, CA, USA) for the first-hand learning opportunities in the area of Silicon photonics. Finally, J.J. Mohr, J.J. Vegas and S.B. Christensen from Mellanox Technologies Denmark are thanked for their constructive discussions and encouragement and I. Tafur and Il-Sug from the Technical University of Denmark for their continued support.

References

[1] N. Eiselt et al., “Real-time 200 Gb/s (4 X 56.25 Gb/s) PAM-4 transmission over 80 km SSMF using quantum-dot laser and silicon ring-modulator,” in 2017 Optical Fiber Communications Conference and Exhibition (OFC), 2017, pp. 1–3.

[2] K. Kurokawa, “Group delay in multimode optical fibers,” Proc. IEEE, vol. 65, no. 8, pp. 1217–1218, Aug. 1977.

[3] D. Marcuse and C. Lin, “Low dispersion single-mode fiber transmission - The question of practical versus theoretical maximum transmission bandwidth,” IEEE J. Quantum Electron., vol. 17, no. 6, pp. 869–878, Jun. 1981.

[4] A. Dochhan et al., “Solutions for 400 Gbit/s Inter Data Center WDM Transmission,” in ECOC 2016; 42nd European Conference on Optical Communication; Proceedings of, 2016, pp. 1–3.

[5] N. Eiselt et al., “First Real-Time 400G PAM-4 Demonstration for Inter-Data Center Transmission over 100 km of SSMF at 1550 nm,” 2016, p. W1K.5.

[6] S. Khatana, “Components for 100G Coherent Pluggable Modules-CFP2,” in Optical Fiber Communication Conference, 2016, p. Th3G–5.

[7] J.-P. Elbers, N. Eiselt, A. Dochhan, D. Rafique, and H. Griesser, “PAM4 vs Coherent for DCI Applications,” in Signal Processing in Photonic Communications, 2017, p. SpTh2D–1.

[8] M. Chagnon, M. Morsy-Osman, and D. V. Plant, “Multi-Dimensional Formats and Transceiver Architectures for Direct Detection With Analysis on Inter-Polarization Phase Modulation,” J. Light. Technol., vol. 35, no. 4, pp. 885–892, Feb. 2017.

[9] K. Kikuchi and S. Kawakami, “Multi-level signaling in the Stokes space and its application to large-capacity optical communications,” Opt. Express, vol. 22, no. 7, p. 7374, Apr. 2014.

[10] D. Che, A. Li, X. Chen, Q. Hu, Y. Wang, and W. Shieh, “Stokes vector direct detection for short-reach optical communication,” Opt. Lett., vol. 39, no. 11, p. 3110, Jun. 2014.

[11] L. G. Kazovsky, “Phase- and polarization-diversity coherent optical techniques,” J. Light. Technol., vol. 7, no. 2, pp. 279–292, Feb. 1989.

[12] F. Frey, R. Elschner, and J. K. Fischer, “Estimation of Trends for Coherent DSP ASIC Power Dissipation for different bitrates and transmission reaches,” in Photonic Networks; 18. ITG-Symposium, 2017, pp. 1–8.

[13] C. R. Doerr, “Silicon photonic integration in telecommunications,” Front. Phys., vol. 3, Aug. 2015.

[14] S. Ghosh, Y. Kawabata, T. Tanemura, and Y. Nakano, “Polarization-analyzing circuit on InP for integrated Stokes vector receiver,” Opt. Express, vol. 25, no. 11, p. 12303, May 2017.

[15] P. Dong, X. Chen, K. Kim, S. Chandrasekhar, Y.-K. Chen, and J. H. Sinsky, “128-Gb/s 100-km transmission with direct detection using silicon photonic Stokes vector receiver and I/Q modulator,” Opt. Express, vol. 24, no. 13, p. 14208, Jun. 2016.

[16] M. Lu, H.-C. Park, E. Bloch, L. A. Johansson, M. J. Rodwell, and L. A. Coldren, “Highly Integrated Homodyne Receiver for Short-reach Coherent Communication,” in Optoelectronic Devices and Integration, 2015, p. OT2A–4.

[17] A. Mizutori, T. Abe, T. Kodama, and M. Koga, “Optical 16-QAM Signal Homodyne Detection by Extracting ±π/4 and ±3π/4-Phase Symbols,” in Optical Fiber Communication Conference, 2017, p. Th4C–6.

[18] Mingzhi Lu et al., “An Integrated 40 Gbit/s Optical Costas Receiver,” J. Light. Technol., vol. 31, no. 13, pp. 2244–2253, Jul. 2013.

[19] P. Velha et al., “Wide-band polarization controller for Si photonic integrated circuits,” Opt. Lett., vol. 41, no. 24, p. 5656, Dec. 2016.

[20] C. R. Doerr, N. K. Fontaine, and L. L. Buhl, “PDM-DQPSK Silicon Receiver With Integrated Monitor and Minimum Number of Controls,” IEEE Photonics Technol. Lett., vol. 24, no. 8, pp. 697–699, Apr. 2012.

[21] M. Ma et al., “Silicon photonic polarization receiver with automated stabilization for arbitrary input polarizations,” in CLEO: Science and Innovations, 2016, p. STu4G–8.

[22] N. Nambath et al., “First demonstration of an all analog adaptive equalizer for coherent DP-QPSK links,” in Optical Fiber Communications Conference and Exhibition (OFC), 2017, 2017, pp. 1–3.

[23] A. Shahpari et al., “Coherent Access: A Review,” J. Light. Technol., vol. 35, no. 4, pp. 1050–1058, Feb. 2017.

[24] J. Tabares, V. Polo, and J. Prat, “Polarization-independent heterodyne DPSK receiver based on 3x3 coupler for cost-effective udWDM-PON,” in 2017 Optical Fiber Communications Conference and Exhibition (OFC), 2017.

[25] S. Echeverri-Chacón et al., “Short range interdatacenter transmission with carrier delivery and remote modulation for 112 Gb/s PM-QPSK signals,” in 19th International Conference on Transparent Optical Networks, 2017.

[26] X. Liu, S. Chandrasekhar, and A. Leven, “Digital self-coherent detection,” Opt. Express, vol. 16, no. 2, pp. 792–803, 2008.

[27] D. Che, Q. Hu, and W. Shieh, “Linearization of Direct Detection Optical Channels Using Self-Coherent Subsystems,” J. Light. Technol., vol. 34, no. 2, pp. 516–524, Jan. 2016.

[28] D. van den Borne, S. Calabro, S. L. Jansen, E. Gottwald, G. D. Khoe, and H. de Waardt, “Differential quadrature phase shift keying with close to homodyne performance based on multi-symbol phase estimation,” pp. 12–12, Jan. 2005.

[29] S. Faralli, K. N. Nguyen, J. D. Peters, D. T. Spencer, D. J. Blumenthal, and J. E. Bowers, “Integrated hybrid Si/InGaAs 50 Gb/s DQPSK receiver,” Opt. Express, vol. 20, no. 18, pp. 19726–19734, Aug. 2012.

[30] J. Klamkin et al., “A 100-Gb/s noncoherent silicon receiver for PDM-DBPSK/DQPSK signals,” Opt. Express, vol. 22, no. 2, p. 2150, Jan. 2014.

[31] P. Velha, S. Faralli, and G. Contestabile, “A Compact Silicon Photonic DQPSK Receiver Based on Microring Filters,” IEEE J. Sel. Top. Quantum Electron., vol. 22, no. 6, pp. 418–424, Nov. 2016.

[32] J. Li et al., “A self-coherent receiver for detection of PolMUX coherent signals,” Opt. Express, vol. 20, no. 19, pp. 21413–21433, Sep. 2012.

[33] X. Chen et al., “218-Gb/s Single-Wavelength, Single-Polarization, Single-Photodiode Transmission Over 125-km of Standard Singlemode Fiber Using Kramers-Kronig Detection,” in Optical Fiber Communication Conference Postdeadline Papers, 2017, p. Th5B.6.

[34] R. Kamran, N. B. Thaker, M. Anghan, N. Nambath, and S. Gupta, “Demonstration of a polarization diversity based SH-QPSK system with CMA-DFE equalizer,” in Wireless and Optical Communication Conference (WOCC), 2017, pp. 1–4.

[35] M. Y. S. Sowailem et al., “100G and 200G single carrier transmission over 2880 and 320 km using an InP IQ modulator and Stokes vector receiver,” Opt. Express, vol. 24, no. 26, p. 30485, Dec. 2016.

[36] Q. Hu, D. Che, Y. Wang, and W. Shieh, “Advanced modulation formats for high-performance short-reach optical interconnects,” Opt. Express, vol. 23, no. 3, p. 3245, Feb. 2015.

[37] C. Doerr et al., “Silicon photonics coherent transceiver in a ball-grid array package,” in Optical Fiber Communications Conference and Exhibition (OFC), 2017, 2017, pp. 1–3.

[38] G. Roelkens et al., “III-V-on-Silicon Photonic Devices for Optical Communication and Sensing,” Photonics, vol. 2, no. 3, pp. 969–1004, Sep. 2015.

Available Resources for Reconfigurable Systems in 5G Networks

Sebastián Rodríguez

Department of Photonics Engineering
Technical University of Denmark
Kongens Lyngby, Denmark 2800
Email: juse@fotonik.dtu.dk

Juan José Vegas Olmos

Mellanox Technologies
Roskilde, Denmark
Email: juanj@mellanox.com

Abstract—**In this paper, the concept of a Radio-over-Fiber based Centralized Radio Access Network is explained and analyzed, in order to identify a set of resources within the network that can be used as a base in the design of reconfigurable systems. This analysis is then used to design a different reconfigurable systems to be implemented as part of the next generation Radio Access Unit. These systems are then implemented and experimentally tested, allowing to demonstrate their operation. The obtained results allow to show the feasibility of the systems and the implementation of a flexible architecture for the next generation of networks.**

Keywords–Mm-wave communications; Optical fiber networks; mobile communication.

I. Introduction

The next generation of networks comes from the ideas of achieving higher capacity, allowing more users in the network and integrating new technologies. These ideas come with different requirements for the networks, in terms of data rates, mobility, latency and spectrum allocation [1].

One step to solve some of these challenges is the use of carriers within the millimeter-wave (mm-wave) frequencies, with the Ka-band (26 GHz to 40 GHz) and the W-band (75 GHz to 110 GHz) being two interesting candidates for the implementation [2]. These higher carrier frequencies allow the use of wider bandwidths in the wireless channels, thus increasing the capacity of the link. The main disadvantage is the high atmospheric attenuation in these frequency ranges. This effect has been addressed by proposing a modification on the Radio Access Network (RAN), consisting of an increase in the number of wireless access points, allowing better coverage in the mm-wave link.

In addition to increasing the number of access points, another big change has also been proposed in the RAN. This change centralizes the processing and operation of the RAN in what is known as the cloud or Centralized RAN (C-RAN). The efforts here are set to simplify the wireless access points, easing their implementation and reducing the implementation (CAPEX) and operation (OPEX) costs of the network [3][4].

Lastly, Radio-over-Fiber (RoF) techniques have been proposed to be the backbone technology of the C-RAN to interconnect the different points of the network and distribute the signals [5][6][7][8]. In this technique, the radio signal is modulated and accommodated to be transmitted in the wireless channel in the source; this signal is then used to modulate a laser, so it can be transmitted directly to the antenna in the optical domain.

In this paper, the architecture of a RoF based C-RAN is analyzed, in order to show the available resources that can be used for the design of reconfigurable networks. Then, some solutions are presented and evaluated, showing the added capabilities of reconfigurable systems within the C-RAN. In Sections II, a general description of the concepts of C-RAN and photonic up-conversion is presented. In Section III, we discuss the available resources in the C-RAN that can be used on the design of flexible systems; Section IV presents some examples of reconfigurable systems in the network. We summarize the discussion and results on Section V.

II. Base technologies

This section explains the two main technologies used as a base for the proposed architectures: the architecture of the C-RAN and photonic up-conversion, used in RoF transmissions to generate the electrical signal in the mm-wave band.

A. The implementation of the C-RAN

The typical architecture of the C-RAN is composed of the Central Office (CO) and the Radio Access Units (RAUs). In this implementation of the RAN, the Base Band Units (BBUs) are taken away from the RAUs, and located in the CO as a virtual BBU (vBBU) pool. This change will simplify the design and implementation of the RAUs and allow the network to centralize the processing in the CO. A typical implementation of the C-RAN is presented in Fig. 1, alongside some examples for different types of access within the network.

The CO will have additional tasks and roles within the network, acting like the gateway between front- and backhaul, being in charge of collecting the signals, process them and redistribute them to each RAU as necessary. This centralization will ease the operation on each RAU and limit the processing points, which in turn will help to reduce the total latency of the system. Moreover, removing the BBU from the RAU will take away most of their complex operations. The signals will be modulated and processed in the CO; and the RAU will accommodate them to be wirelessly transmitted. Therefore, the RAUs will work as interfaces between the wireless and the optical networks.

The simplification of the RAU comes with an additional advantage under the context of the implementation of mm-wave band links. Since the RAUs are the wireless access points of the network, their simplification will ease and reduce costs for their implementation and the desification process that was proposed to increase the total coverage of the network. In addition, the simplified design of the RAUs, will also allow

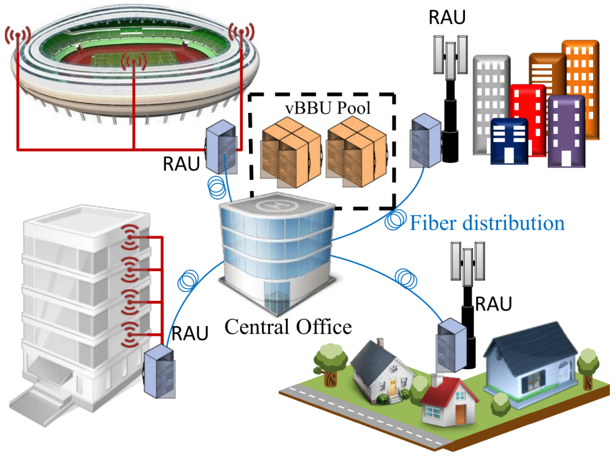


Figure 1. Typical implementation of the C-RAN. The heavy processing is performed in the vBBU pool located in the Central Office. The signals are centralized in the Central Office and then distributed to the RAUs. RAU: Radio Access Unit; vBBU: virtual Base Band Unit.

the implementation of new architectures within the RAN, as is the case of distributed antennas, to improve the connection in indoor and wide outdoor areas.

B. Photonic Up-conversion

The photonic up-conversion is a method to generate the mm-wave signals using the characteristics of optical devices as an advantage. In this process, the signal is generated through direct heterodyning in a wideband Photodiode (PD) [9]. This method takes use of two optical signals: one that contains the information ($s(t)$) modulated in a RoF signal that comes from the central office ($E_s(t)$), with an angular frequency ω_s; and one that is going to be used as an optical Local Oscillator (LO, $E_{LO}(t)$) at angular frequency ω_{LO}. The two signals are then expressed as:

$$E_s(t) = \sqrt{P_s}s(t) \cdot e_s \exp[-j(\omega_s t + \phi_s)] \quad (1)$$

$$E_{LO}(t) = \sqrt{P_{LO}}e_{LO} \exp[-j(\omega_{LO} t + \phi_{LO})] \quad (2)$$

where P_s, ϕ_s e_s represent respectively the power, phase and polarization unit vectors for the modulated signal; and P_{LO}, ϕ_{LO} and e_{LO} are the values for the LO. The electrical signal given at the PD will be composed of two components, one baseband component and one RF signal with a carrier of $\omega_{RF} = |\omega_s - \omega_{LO}|$. The baseband component is filtered by the limited bandpass bandwidth of the antenna. The final transmitted RF signal can be described as

$$E_{RF}(t) = 2\sqrt{P_s \cdot P_{LO}} \cdot s(t) \cdot e_s e_{LO} \cdot \cos[\omega_{RF} t + \phi_{RF}(t)]. \quad (3)$$

III. Available resources in the 5G generation access network

In the CO, the data will be processed in a BBU and it will be electrically modulated on an Intermediate Frequency (IF). The signal is then transformed into the optical domain and transported through the optical channel to the RAU using RoF techniques. In the RAU, the photonic up-conversion process (explained in Section II) takes place, transforming the signal back to the electrical domain for transmission through the wireless channel. The connection between the CO and the different RAUs is summarized in Fig. 2. Generally, in these applications, the PD (Optical/Electrical interface) and the Power Amplifier (PA) are integrated with the antenna. Therefore, the RAU will transport the signal to the antenna directly in the optical domain.

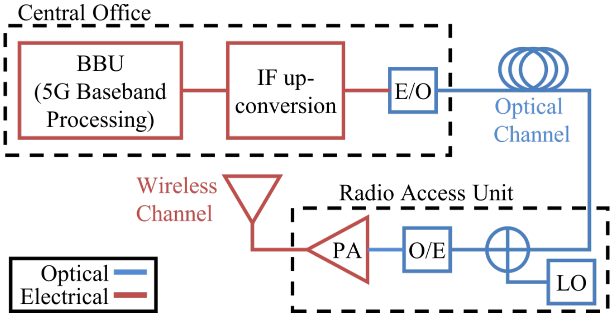


Figure 2. Simplified scheme of the signal path of the C-RAN. Composed by the Central Office, the optical channel, the Radio Access Unit and the wireless channel. BBU: Base Band Unit; IF: Intermediate frequency; E/O: Electrical to optical conversion; LO: Local Oscillator; O/E: Optical to electrical conversion; PA: Power amplifier.

In both the optical and wireless channels, there are different kinds of resources and applications that the network can use to achieve a better and more efficient transmission of the information. The following subsections provide a short review of some techniques previously proposed.

A. Optical channel

The following techniques make use of optical technologies or devices to add extra functionality to the network. We will focus on three main techniques:

1) Wavelength: Within the optical channel, one of the most used resources is the wavelength of the optical carrier. In Wavelength Division Multiplexing (WDM), different signals are modulated in different carrier wavelengths, with a wavelenght separation predefined by the ITU standard [7][8][10]. All the RAUs will receive all the WDM signals and dynamically select one according a control signal from the CO.

2) Optical switches: Some optical channel use optical switches to redirect the signal to different points of the network [10][11][12]. In these implementations, the data is distributed in packets and signaling information is sent along with it, either in form of a label on the packet or a synchronization order from the CO. The RAU will then redirect the signal to different fibers that lead to different antennas, either in the same location or in distributed arrays of antennas.

3) Multicore Fibers: The last technique that has been proposed in these area, corresponds to the use of Multicore Fibers (MCF) in the implementation of the access networks [6]. In this technique, the different cores of the MCF are used to transmit different signals through the network. Each core is used as a different channel, which can be either reserved for up-link or down-link. Therefore, both the RAU and the CO can select in which core to transmit and in which to receive an optical signal.

B. Electrical channel

The techniques implemented in the electrical domain focus on the use of the RF spectrum or in modifications within the antenna to give extra tunability to the network. There are mainly three techniques to explore:

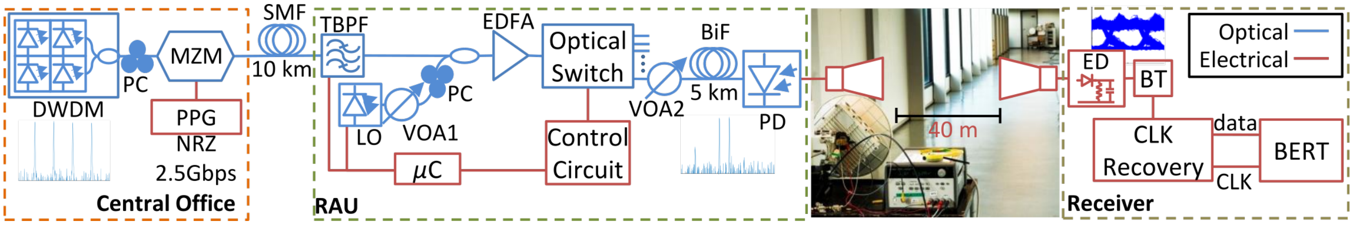


Figure 3. Implemented experimental setup to test the reconfigurable systems. DWDM: Dense wavelength multiplexion; PPG: Pulse pattern generator; NRZ: Non-return-to-zero; PC: Polarization controller; MZM: Mach-Zehnder modulator; SMF: Single mode fiber; RAU: Radio access network; TBPF: Tunable band pass filter; LO: Local Oscillator; μC: Micro-controller; VOA: Variable optical attenuator; EDFA: Erbium-doped fiber amplifier; BiF: Bend-insensitive fiber; PD: Photodiode; ED: Envelope detector; BT: Bias tee; CLK: Clock; BERT: Bit-error-rate tester..

1) Carrier frequency: Similar to the optical domain, the first technique consist of the use of multiple carrier frequencies during the transmission of the signal. In this case, the LO used within the system can be tuned, allowing to the RAU to change the carrier frequency as necessary [7][13].

2) Beam steering: In order to compensate for the atmospheric attenuation, it is desired to use antennas with high gain. However, with a higher gain, the antenna becomes more directive, which results in a low coverage area. The beam steering techniques aim to modify the direction pattern of these antennas, allowing them to change the direction of the main transmission lobe so they can achieve better coverage [14]. In this technique, the network will get feedback on the status of the communication link from the user terminal and adjust the radiation pattern of the transmitting antenna in order to enhance the power on each wireless link connected to the network.

3) Reconfigurable antennas: This technique modifies the antenna's pattern or operation to give additional functions to the system, that not necessaryly focus on the direction of the main lobe.This can be as is in the case of the Optically Controlled Reconfigurable and Multiband Slotted waveguide Antenna Array (OCRAA) [15] or the T-shaped antenna [16]. The former is an antenna which implements a silicon piece that acts as a switch in the antenna, modifying the radiation pattern by turning the antenna ”on” and ”off”. The latter refers is an antenna that uses variable resistors to modify the resonant frequency of the antenna.

IV. Reconfigurable Systems in the Access Network

In this section, we collect some proposed solutions and their results, to design reconfigurable RAUs. All of them are based on the same principle and same base architecture, as presented in Section III. A more detailed configuration is presented in Fig. 3, based on the proposed system presented in [7]. This architecture gives a general overview of the general components for the implementation of wireless links enhanced with the implementation of RoF.

In the Central office, the DWDM signal is generated with a set of equally distanced lasers. Then, a Mach-Zehnder Modulator (MZM) is used to modulate these lasers with a Non-Return to Zero (NRZ) Pseudo-Random Bit Sequence (PRBS) of $2^{15}-1$ bits at a rate of $2.5\,\mathrm{Gbps}$ coming from a Pulse Pattern Generator (PPG).

The signal propagates through a Standard Single Mode Fiber (SSMF) to reach the RAU. In the RAU, the reconfigurable systems will be installed and a LO will be added to the signal, before reaching the wideband PD (in this case with a bandwidth of $90\,\mathrm{GHz}$), which will convert the signal to the electrical domain. The electrical signal will then be wirelessly transmitted by a high gain antenna.

The receiver will collect the signal using a second high gain antenna and demodulate it. In this scenario, high order modulations of are not explored, meaning that this process can be performed using an Envelope Detector (ED). The signal is then recorded with a Digital Storage Oscilloscope (DSO) or analyzed in real time with a Bit Error Rate Tester (BERT).

The following subsections, will discuss the design of the different reconfigurable subsystems and show some of the obtained results of their operation.

A. Wavelength selection

The wavelength selection system, was designed using a tunable bandpass optical system. The central frequency of the system is controlled by a micro-controller (μC) connected to the network. Once the μC receives the order to change the value of the filter, it uses a Digital-to-Analog Converter (DAC) connected to the filter to move the passband.

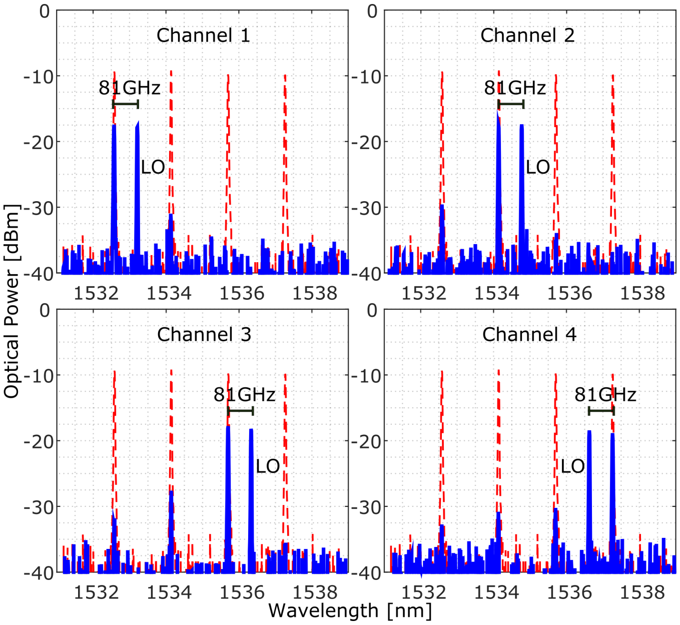


Figure 4. Reference DWDM signal (red) with the selected channel and corresponding optical local oscillator.

Four modulated optical channels with a spacing of $100\,\mathrm{GHz}$ between carrier wavelengths were used to test the system. Fig. 4 shows the spectra of each channel after being selected with the filter and the added local oscillator. The original DWDM is shown as reference in red in the background of the spectra. From the results, we can see that each signal was correctly extracted but that there are still some residues of the other channels, due to the slope and maximum attenuation of the frequency response of the filter. Two types of channels can be identified: the corner channels, corresponding to channels 1 and 4; and the center channels, channels 2 and 3. The main difference between these channels is that the center channels will have interference coming from the two adjacent channels, making their performance different than the one for the corner channels.

The measured BER traces can be found in Fig. 5(a). It can be noted that all channels present error-free results. As expected the slope of the traces for the corner channels follows the same performance, but the center channels are affected by the interchannel interference and present a different performance. The sensitivity is around $-4.4\,\mathrm{dBm}$ for channels 1, 2 and 3, and $-5.6\,\mathrm{dBm}$ for channel 4, after comparing to the Forward Error Correction (FEC) limit, corresponding to an overhead of 7%,.

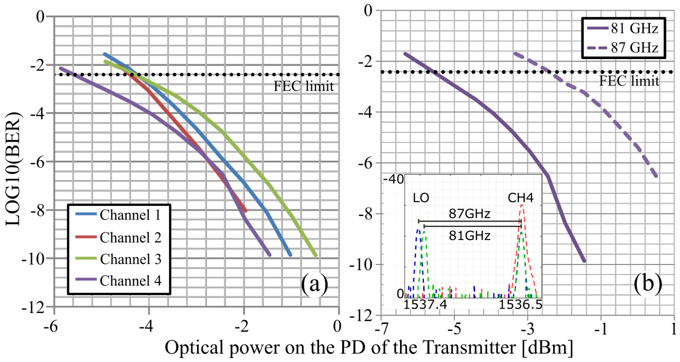


Figure 5. Measured BER vs optical power at the PD located in the RAU: (a) For the different wavelength channels; (b) for the different mm-wave Carrier frequencies.

B. Carrier Frequency selection

This system was designed by employing a tunable laser as LO. The tunable laser is controlled by the same μC as before. Once the wavelength of the LO laser changes, the output carrier frequency of the system changes in a similar way, as explained in Section II.

Fig. 5(b) shows both the optical spectra for the two different LOs and the measured BER of the two systems. In the demonstration the two resulting electrical carrier frequencies were 81 and $87\,\mathrm{GHz}$ as shown in the optical spectrum. The BER traces show that at $87\,\mathrm{GHz}$ the performance of the systems deteriorates significantly. The sensitivity of the system, compared to the FEC limit, moves from $-5.6\,\mathrm{dBm}$ to $-2.6\,\mathrm{dBm}$ for the higher frequency. This $3\,\mathrm{dB}$ difference corresponds not only to added attenuation on the channel, but the change of the slope shows that there is an additional effect. The performance is also affected by the cut-off frequency of $90\,\mathrm{GHz}$ of the PD.

C. Optical Switches

For this application, the solution is based on the design presented in [11]. In this case, a similar topology as in Fig. 3 was used to generate wireless packets in the Ka-band. This implementation consisted on the use of an optical switch to transfer the signal to different antennas in the RAU. The optical switch is controlled by a synchronization signal given by the CO; in the experiment, this signal is generated directly in the PPG.

Fig. 6 shows the operation of the system, in terms of the measured BER and the recorded times slots of the packets. The BER traces were captured for the Back to Back (B2B) case and after the wireless transmission. In both cases, the performance presents similar slope, the main difference between the two being the added attenuation due to the wireless channel. Additionally, Fig. 6 also shows the switching process of the received packets divided in time slots. In this figure, a packet can be observed in a time slot with a blank space used as safeguard for the transition. The different lines show the process of switching between a different number of outputs, demonstrate the correct transmission of the packets.

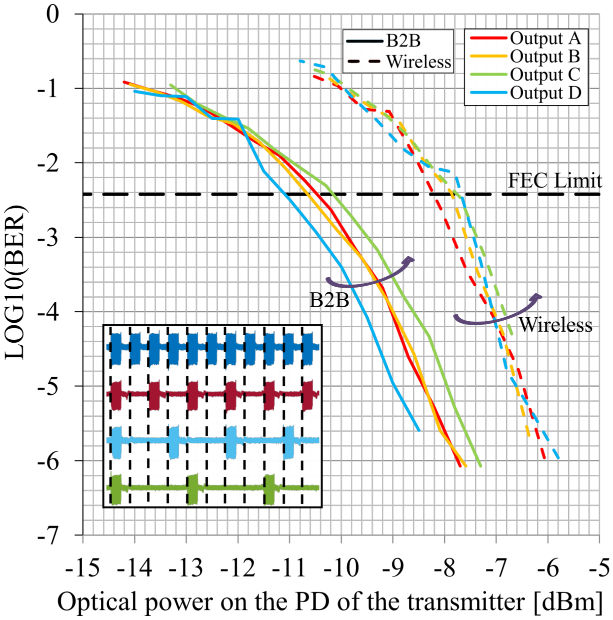


Figure 6. Measured BER vs optical power at the PD located in the RAU for the implementation using optical switches. In the side, the generated packet in time slots, with each line showing the effect of transmitting the packet to one, two three and four active outputs.

D. Reconfigurable Antennas

Fig. 7 shows the device presented and tested in [15]. This device consist of a slotted waveguide antenna that has a silicon switch in its slots. Once the silicon pieces are illuminated (”on” state), the radiation pattern of the antenna changes, creating two states of operation for the device. When the device is in the ”on” configuration, the gain will increase approximate of $9\,\mathrm{dB}$. This functionality allows the use of an extra boost of power to have higher coverage in indoor applications.

E. Beam steering

The advantage of beam steering is the capacity to redirect the main lobe of an antenna array. This can be used to enhance the coverage of a RAU or to use it as a switch between different receivers, thereby creating a dynamic wireless bridge. One example is shown in Fig. 8, based on the implementation proposed in [17]. Here two similar transmitters where placed side by side and a receiver antenna with a mechanical beam

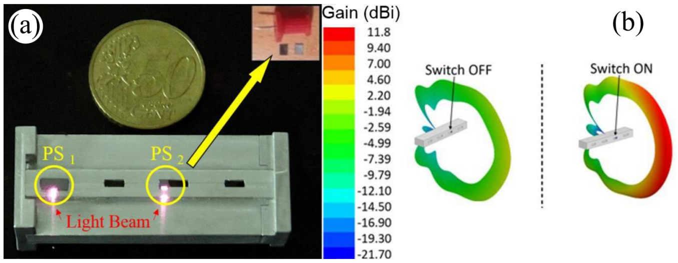


Figure 7. Optically reconfigurable antenna: (a) fabricated device and (b) Operation

steering platform was used to switch between them, as shown in 8(b) and (c).

The BER traces in Fig. 8(a) are the measurements of the received signal for each transmitter in two cases: first in the case that only one of the transmitters is active at the time of measurement used as reference to see the channel interference; and the other case to analyze the system with the two transmitters active.

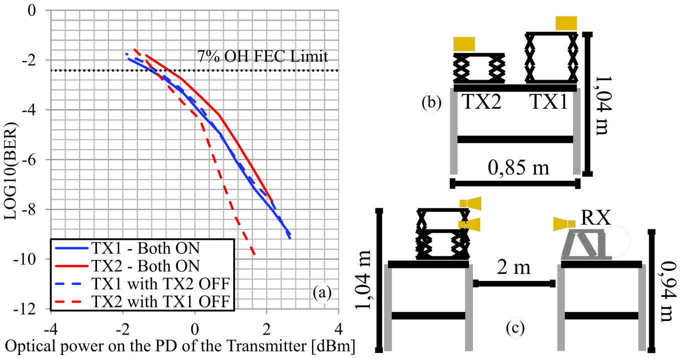


Figure 8. Example with mechanical beam steering: (a) Measured BER traces vs power on the PDs, (b) Front view of implementation (c) Side view of implementation

To generate the two signals, the RoF signal was divided in two paths; each path was used to feed a different PD and then go through the amplifiers and the antennas. In one of the paths, an extra waveguide section was used to generate diversity between the signals. From the curves it be observed the effects of the interference in TX2 being higher than in TX1. This is due imbalances in the power of both transmitters due to the different components in the paths. Nevertheless, all the traces seem to join for a sensitivity around $-1.2\,\mathrm{dBm}$ and present error-free transmissions.

V. Conclusion

In this paper, an overview on the architecture of the RoF based C-RAN and the basic elements that compose it was presented. The architecture was simplified and analyzed in order to provide a insights on reconfigurability within the access network. This analysis was expanded with a set of experimental results for different reconfigurable systems in the network. The presented systems serve as proof of concept for techniques as dynamic wavelength and electrical frequency carrier selection, optical switches, beam steering and reconfigurable antennas. In each test, the results presented values below the FEC limit for a 7% of overhead, showing the feasibility of the solutions. Nevertheless, since the C-RAN requires that the RAUs to be cost-efficient and easy to deploy, there is still work to consider in the integration of the diferent components and the design of the control information within the newtork, ir order to implement the RAU as part of a software defined network.

In overall, the presented solutions allow to show the capability of a C-RAN architecture, completely controlled by the central office, providing some insights on a reconfigurable optical and electrical system for the next generation of networks and leaving the discussion open for the introduction of new designs and new components for the network.

Acknowledgment

This work was supported by the Marie Skodowska-Curie Innovative Training Network FiWiN5G supported by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 642355. The authors will like to thank also Idelfonso Tafur Monroy, Alvaro Morales Vicente of the Eindhoven University of Technology (TU/e) and Igor Feliciano Da Costa of the National Institute of Telecomunications (Inatel) for their support, work and ideas on the implementations of the original implemented systems.

References

[1] Dahlman et al., “5G wireless access: requirements and realization,” IEEE Communications Magazine, vol. 52, no. 12, dec 2014, pp. 42–47.

[2] Rappaport et al., “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!” IEEE Access, vol. 1, 2013, pp. 335–349.

[3] Ranaweera, Wong, Nirmalathas, Jayasundara, and Lim, “5G C-RAN architecture: A comparison of multiple optical fronthaul networks,” in 2017 International Conference on Optical Network Design and Modeling (ONDM), 2017, pp. 1–6.

[4] I, Huang, Duan, Cui, Jiang, and Li, “Recent Progress on C-RAN Centralization and Cloudification,” IEEE Access, vol. 2, 2014, pp. 1030–1039.

[5] Kitayama, Kuri, Olmos, and Toda, “Fiber-wireless networks and radio-over-fibre technique,” in 2008 Conference on Lasers and Electro-Optics and 2008 Conference on Quantum Electronics and Laser Science, May 2008, pp. 1–2.

[6] Galve, Gasulla, Sales, and Capmany, “Reconfigurable Radio Access Networks Using Multicore Fibers,” IEEE Journal of Quantum Electronics, vol. 52, no. 1, jan 2016, pp. 1–7.

[7] Rodriguez, Morales, Rommel, Vegas Olmos, and Tafur Monroy, “Real-time Measurements of an Optical Reconfigurable Radio Access Unit for 5G Wireless Access Networks,” in Optical Fiber Communication Conference. Washington, D.C.: OSA, 2017, p. W1C.3.

[8] Kitayama, Kuri, Olmos, and Toda, “Fiber-wireless networks and radio-over-fibre technique,” in 2008 Conference on Lasers and Electro-Optics and 2008 Conference on Quantum Electronics and Laser Science, May 2008, pp. 1–2.

[9] Lebedev et al., “Feasibility study and experimental verification of simplified fiber-supported 60-ghz picocell mobile backhaul links,” IEEE Photonics Journal, vol. 5, no. 4, Aug 2013, pp. 7 200 913–7 200 913.

[10] Rodríguez, Rommel, Olmos, and Monroy, “Reconfigurable radio access unit to dynamically distribute W-band signals in 5G wireless access networks,” Optical Switching and Networking, vol. 24, 2017, pp. 21–24.

[11] Rodriguez, Madsen, Monroy, and Olmos, “Dynamic optical fiber delivery of ka-band packet transmissions for wireless access networks,” in 2017 International Conference on Optical Network Design and Modeling (ONDM), May 2017, pp. 1–4.

[12] Liu, Zhang, Zhu, Wang, Cheng, and Chang, “A Novel Multi-Service Small-Cell Cloud Radio Access Network for Mobile Backhaul and Computing Based on Radio-Over-Fiber Technologies,” Journal of Lightwave Technology, vol. 31, no. 17, sep 2013, pp. 2869–2875.

[13] Shams, Fice, Gonzalez-Guerrero, Renaud, Dijk, and Seeds, “Sub-THz Wireless Over Fiber for Frequency Band 220280 GHz,” Journal of Lightwave Technology, vol. 34, no. 20, oct 2016, pp. 4786–4793. [Online]. Available: http://ieeexplore.ieee.org/document/7460176/

[14] Razavizadeh, Ahn, and Lee, "Three-Dimensional Beamforming: A new enabling technology for 5G wireless networks," IEEE Signal Processing Magazine, vol. 31, no. 6, nov 2014, pp. 94–101.

[15] Costa et al., "Optically controlled reconfigurable antenna for 5G future broadband cellular communication networks," Journal of Microwaves, Optoelectronics and Electromagnetic Applications, vol. 16, no. 1, mar 2017, pp. 208–217.

[16] Jilani, Abbas, Esselle, and Alomainy, "Millimeter-wave frequency reconfigurable T-shaped antenna for 5G networks," in 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, oct 2015, pp. 100–102.

[17] Rodríguez, Morales, Gallardo, Olmos, and Monroy, "Real-time 2.5 Gbit/s spatial circuit switching on W-band wireless links," Optical Engineering, vol. 56, no. 2, 2017, p. 26104.

Evaluation of SDN Enabled Data Center Networks Based on High Radix Topologies

Bogdan Andrus[(1)(4)], Victor Mehmeri[(1)], Achim Autenrieth[(4)]

(1) Department of Photonics Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark.
email: bogan@fotonik.dtu.dk
(4) ADVA Optical Networking SE, Fraunhoferstr. 9a, 2152 Martinsried / München. Germany.

Juan José Vegas Olmos[(1)(2)], Idelfonso Tafur Monroy[(1)(3)]

(2) Mellanox Technologies, Denmark.
(3) Department of Electrical Engineering, Technical University of Eindhoven, Netherlands.

***Abstract*— The relevance of interconnects for large future datacenters and supercomputers is expanding as new technologies like Internet of things (IoT), virtual currency mining, High Performance Computing (HPC) and exa-clouds integrate further into data communication systems. The Data Center Network Layer is the workhorse that manages some of the most important business points by connecting the servers between them and delivering high performance to users. Evolution of the networking layer has seen, in addition to improvements of individual link speeds from 10Gb/s to 40Gb/s and even 100Gb/s and beyond, quite important changes in the topological design. Traffic intensive server-centric networks and high performance computing tasks are pushing a shift from the conventional Layer 2 oriented fat tree architectures with multiples tiers towards clos networks and other highly interconnected matrices. Optimal performance and reliability perquisites imposed on the network cannot be fully achieved by solely changing the topological design. A software-centric control of the network enables the use of additional redundant paths not only for increased performance but also reliability concerns. By decoupling the network control from individual devices and centralizing the network intelligence inside a Software Defined Network (SDN) controller, dynamic workloads can easily be accommodated with the deployment of custom modules or applications for traffic management. In this paper, we focus on the innovations for next generation data center networks from a twofold perspective. On the one hand, we evaluate the applicability of new potential interconnection schemes like torus, hypercube, fat tree and jellyfish in regard to identified key metrics such as performance, complexity, cost, scalability and redundancy. Our evaluation comprises of a mathematical interpretation of the graphs with a focus on the abstract metrics (e.g., bisection bandwidth, diameter, port density, granularity etc.) followed by a simulation of the scalable networks in a virtual environment and subject them to various traffic patterns. On the other hand, we introduce an emulated SDN test framework, which decouples the control plane from the interconnection nodes and gives a centralized view of the topology to a controller handling the routing of the internal workflows for the data center. With the use of our SDN enabled testbed we demonstrate and highlight the clear superior performance gain of centralizing the network intelligence inside a software controller, which allows us to apply a custom routing algorithm.**

Keywords- SDN; Data Center topologies; Torus; Hypercube; Jelly Fish; Fat Tree.

I. INTRODUCTION

New technological trends like IoT, virtual currency mining, High Performance Computing (HPC), exa-clouds integrate further and further towards data communication systems making the role of interconnects more important than ever before. The current global evolution of data center traffic is predicted to reach an annual rate of 15.3 zettabytes (ZB) - with a monthly rate of 1.3 ZB - by the end of 2020 [1]. This prediction translates into tripling traffic demands over a period of 5 years spanning from 2015 to 2020 with a compound annual rate of 27%. The distribution of data center related traffic regards the majority of connections established inside the data center with a quota of 77% for server-to-server communication. Major factors influencing such patterns are identified in distributed computing/processing as well as reliable and fast migration of sizable volumes of data across vast domains of physical servers. Such circumstances highlight the importance of the network topology in the process of designing data centers, on account of the fundamental limits (e.g., cost and performance) imposed by the chosen interconnection graph.

Meanwhile, Big Data and Cloud applications, which are subject to exponential growth rates are pressuring Data Center enterprises to drastically improve their infrastructure not just to meet the increased bandwidth demand but also additional QoS perquisites related to certain applications and services. As a consequence, such priorities are placing the boost for network capacity in the top research directions weather they focus on enhancing individual link capacity to 40Gb/s, to 100Gb/s or beyond or by deploying special network topologies and routing structures [2]-[5]. An

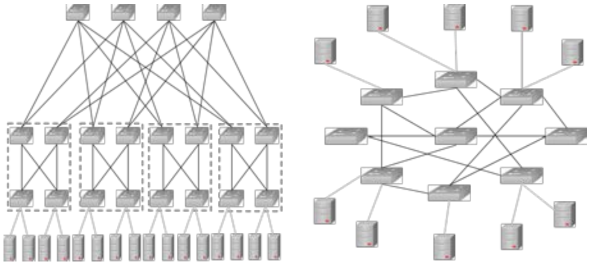


Fig. 1. (a) - Fat Tree (left) and (b) - JellyFish (right) topologies

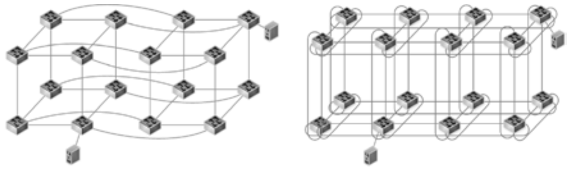


Fig. 2. (a) - Hypercube (left) and (b) - Torus (right) topologies

interconnection matrix is not only an important component in a server-centric Data Center construct but also crucial element for efficient on-chip networks [6]. For this reason, topologies originally adopted in parallel computing platforms and on-chip networks, that present higher interconnectivity between nodes, have gained more and more traction in Data Center network deployments [7].

As identified in [8], one major constraint not tackled by these graphs is the issue of granular scalability or the capability of adding a flexible number of servers or increasing capacity while maintaining structural properties. The number of redundant links, but most importantly their placement in the overall graph raises the network's capability to experience local failures without major impact on operations. The reliability (fault tolerance) of such constructs entails a compromise in terms of additional underutilized spare links or a disproportionate increase of hop count in link failure situations. Furthermore, implications connected to the manual configuration of such networks and potentially complex routing mechanisms not only translate into additional costs related to Capital Expenditure (Capex) but also Operational Expenditure (Opex).

Software Defined Networking (SDN) approach to network management and configuration seeks to bring the flexible programmability needed by real time applications and services, which can considerably reduce the set-up time. One major benefit from adopting an SDN framework relates to lowering Capex and Opex costs. Firstly, low-cost white box switches can be developed by detaching the control plane functionality from all network devices. This step is followed by centralizing their behavior inside a software controller responsible for the management and control operation of the entire network. As such, minimizing the expenses is achieved by replacing the large number of nodes that are capable of supporting complex path computation algorithms with white-box switches that provide fewer features but present a more flexible and reconfigurable alternative. Maintaining a general overview and supervision of every network device inside a dynamic software controller can facilitate overall network management and configuration. Therefore, by automating the manual operations of managing and configuring every network device (also required when scaling such intricate interconnections), Opex oriented costs can also be lowered.

The contributions of this paper can be divided into several sections. In Section 2, we extend our previously presented mathematical analysis [9] of high radix topologies (e.g., Torus, Hypercube) with regard to indications on performance, cost, latency of implementation for new topologies (e.g., Fat Tree, Jellyfish). Section 3 highlights the results and behavior of scaling such topologies in a simulation environment (e.g., [10]) using a random traffic pattern. In Section 4, we present the evaluation testbed for measuring the performance (e.g., network throughput, delay, jitter, and loss rate) of an SDN implementation employing each topology against conventional Spanning Tree Protocol deployment based on our previously published works [11, 12]. Finally, in the Section 5, we present and discuss the results obtained from the simulation and emulation testbeds.

II. Background on Network Topologies

A key component in the performance of a Data Center architecture network is the topology. The impact of a topology is not only significant for the global network ratio of performance vs. cost but also for the failure resiliency aspect. Traversing nodes and links incurs energy, and since the number of hops for the various paths is affected by the interconnection implementation, an important role in the energy consumption can also be easily identified.

A hypercube graph, Figure 2(a), is an n-dimensional generalization of a cube also called n-cube and comprises of 2n nodes. One main characteristic is the high connectivity and small diameter however, not very easy to scale due to complexity. A torus topology can be visualized as a three-dimensional mesh in the shape of a rectangular prism with all the nodes on each face having an additional connection to the corresponding nodes on the opposite face, as illustrated in Figure 2(b). Torus based networks are usually employed in top performing supercomputers due to their high radix, relatively low cost and reduced diameter compared to mesh. Another widely deployed Data Center topology is Fat-Tree, Figure 1(a), which is capable of delivering high bisection bandwidth due to its path multiplicity and maintain a low and constant diameter if the number of layers remains constant with scaling. On the other hand, Jellyfish, Figure 1(b), a random graph and multipath based topology, has been proven to be more cost-efficient than a Fat-Tree using identical devices, providing support for 25% more servers running at full capacity [13]. Furthermore, Jellyfish graph provides an attractive solution for a more granular expansion and allows heterogeneity in switch port count, a desired advantage in terms of flexibility.

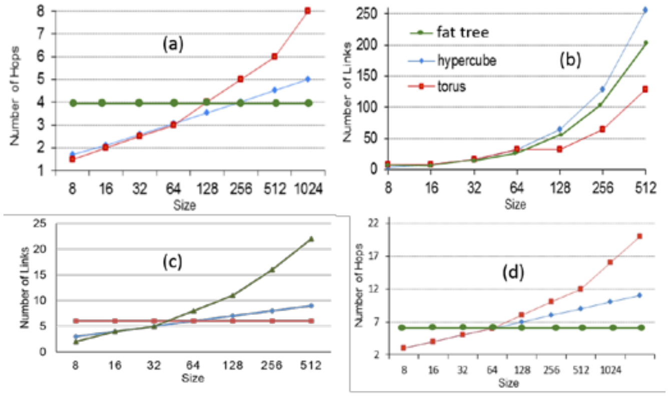


Fig. 3. Average Distance (a), Bisection Bandwidth (b), Node Order (c), Diameter (d)

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-591-3

Some relevant mathematical parameters by which topologies can be characterized and compared in a preliminary network design stage have been identified in previous research [14]. Such parameters can serve as the building blocks for clarifying certain prospects, in relation to scalability for each network but also in relation to each other.

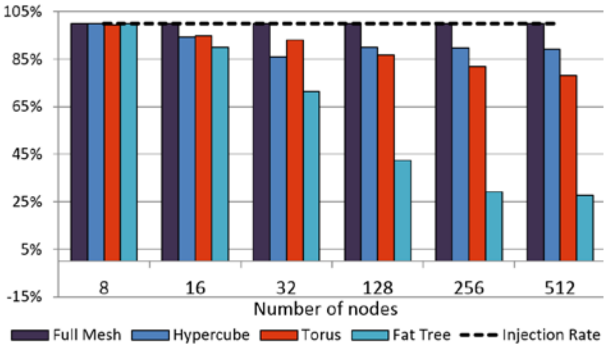


Fig. 4. Average throughput per node

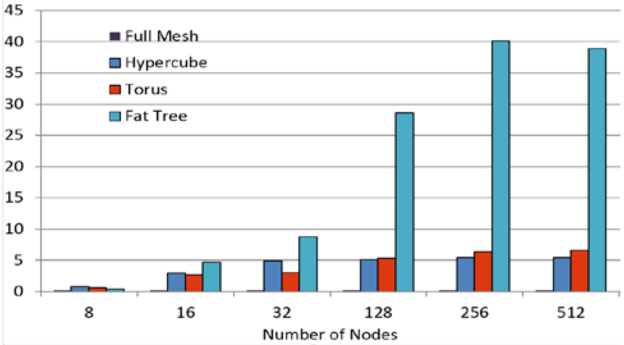


Fig. 5. Average delay per flow

Bisection bandwidth (Figure 3(b)), the bandwidth sum of all the links across a cut through the middle of the network, gives the link density and bandwidth indications that can be achieved by a certain implementation strategy. Even though the order of growth is similar in torus, hypercube and fat tree, the bisection reaches larger values for hypercube and fat tree. This asset of superior bandwidth comes with a setback related to cost and complexity of wiring on the hypercube side. However, this difference is almost inexistent for 64 nodes and below.

The longest path between any two nodes calculated on the shortest path tree (diameter), is arguably a sign of packet latency, as seen in Figure 3(d). Another similar guideline is average distance in the network, highlighted in Figure 3 (a), which also supports latency in relation to communication patterns. While a similar tendency is observed for both hypercube and torus having a logarithmic and a linear increase, respectively, a 3 layer (e.g., edge, aggregation, core) fat tree maintains a constant diameter if the number of layers is unchanged when expanding.

Node order represents the number of interconnection links (ports) required for each switch and relates to network throughput, however, implementing a system with a high node order implies an increased execution complexity cost. Like in the previous cases, for graphs up to 64 nodes the differences are relatively small between interconnections. By observing the evolution of this parameter, Figure 3(c), we note that the torus network can scale up and maintain a constant node degree. However, the fat tree is characterized by a higher increase rate in switch-to-switch port count when graphs scale.

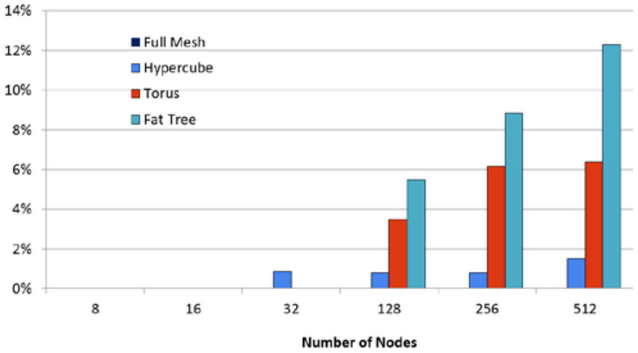


Fig. 6. Average loss rate per flow

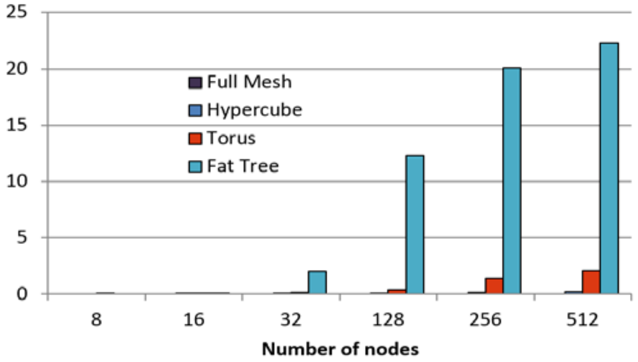


Fig. 7. Average jitter per flow

The next section presents the results of simulating the expansion of the topologies with regard to key performance metrics: throughput, latency, lost rate and jitter.

III. Network Performance Simulations

The following simulations try to provide a comprehensive interpretation of the highly interconnectable networks assessed. In this scope, when setting up our simulation models, we are considering topology characteristics, communication pattern and the amount of data injected in the network as being the most relevant components for our scenarios.

In order to simulate the proposed scenarios, we used NS3, an open source discreet event network simulator capable of supporting network performance measurements related to throughput, delay, number of lost packets, jitter etc.

As in our previous evaluation [9], we have selected a shortest path based routing algorithm as opposed to a Spanning Tree Protocol implementation in order to evaluate the real potential of the topologies without blocking redundant ports for communication. The networks are subject to a uniformly distributed random traffic model that is widely accepted and is unbiased towards various topologies (e.g., fat tree behaves better under localized traffic patterns between neighbors in the same pod/cluster).

Besides the traffic pattern, the amount of traffic that will be injected in the interconnection network also plays an important role. Above a certain threshold the network begins to saturate, the ratio between throughput and injection rate starts to drop and the efficiency decreases. The saturation limit for hypercube and torus is 60% and 40%, respectively [15]. Therefore, we configure the input traffic level to be at 30% the maximum link capacity, below the saturation limit. Due to the random traffic pattern selection the results were averaged from a runtime of 30 seconds during which application based connection flows would be established between randomly paired hosts in the network.

As expected from the initial abstract metric analysis, we observe from Figure 4 that, even though the throughput per flow is declining when the networks scale from 8 nodes up to 512, the hypercube topology presents the most consistent behavior. A decrease of approximatively 5% in hypercube compared to 15% corresponding with the torus or a drastic 70% decrease in fat tree performance is measured.

Similar behavior occurs in the investigation of the delay (Figure 5) where even though the torus and hypercube demonstrate a similar delay, the 3-layer fat tree does not indicate well performance on scaling with an average of up to 40 ms per flow in 512 switched network. Same trend is observed when analyzing the rate of lost packets (Figure 6) and jitter (Figure 7) where hypercube and torus outperform the fat tree based network.

We have simulated and tested the performance of a full mesh topology (all switches have direct connections to every other switch in the network) under the same conditions as the other interconnects in order to perform a sanity check for the setup. First, this serves as a verification for all the configuration parameters employed, data rates, individual link delays, simulation time etc. Secondly, the test demonstrates the confidence in the results, which were not affected by hardware processing power when the topologies scaled from 8 to 512 nodes.

We can conclude from this section that, even though the cost indications and performance characteristics are similar for all studied topologies with nodes under a 64 count, clear superior operations are displayed by the hypercube followed by the torus.

IV. SDN For Data Center Interconects

Applying high radix topology designs into large data centers by employing current routing or switching technologies encounters numerous obstacles. A Layer 2 topology builds upon high density port switches that can process packets at line speed therefore, this implies less configuration and administration overhead. However, such a solution does not scale well due to the limitations of a flat topology and restrictions to a broadcast domain.

A routing based deployment can segment the broadcast domains, use existing performant routing protocols and present better scaling properties. Nevertheless, this comes at the expense of additional delays incurred by additional packet processing times in routers, which are not only slower but are also more expensive and require more complex administration [16]. In data centers, generally we see a compromise based on a combination of the two but also replacing network elements with expensive multilayer switches is an available option. However, SDN can truly get the best of both scenarios by giving routing capabilities to lower cost, white-box switches and automate administration operations with a topology manager module in the controller.

SDN adoption has raised many concerns about its impact on performance and scalability mainly due to the fundamental aspect of decoupling the control plane from the data plane. Ideas that a centralized controller is unlikely to scale as the network grows has led to some reluctance and certain expectations that some failure would occur when the number of incoming requests increases over supported limits. These assumptions can generally be sourced to the misconception that SDN implies the use of one physically centralized controller. Architectures involving a distributed control plane are, however, a valid way to construct a Software Defined Network and address the scalability issue, while also providing control plane resilience. Such solutions have already been demonstrated in projects like Onix and HyperFlow [17][18]. Yeganeh argues in [19], that there is no underlying bottleneck to SDN scalability, such a concern is not fundamentally unique to SDN. Even though a distributed SDN architecture would incur similar manageability problems as in a non-SDN approach, it would still be significantly easier to manage compared to having multiple heterogeneous switches running autonomous, vendor-specific applications.

V. Performance Evaluation of SDN DC Topologies

With the scope of evaluating and comparing SDN versus STP implementations on Hypercube, Torus and Jellyfish topologies, we used Mininet, a network emulation software that uses process-based virtualization to run multiple virtual OpenFlow switches and hosts on a same physical machine.

The SDN network controller of choice was Floodlight. The integrated topology and forwarding module enable the efficient use of the high number of redundant paths for networks like torus and hypercube and calculates connections based on shortest path between each pair of source and destination. However, link contention still occurs, influencing performance as seen in the results.

We have employed Iperf, a linux networking tool in order to measure network performance characteristics. We have focused on the same performance metrics identified as the most common attributes for network characterization in academic research, similar to our previous simulation scenarios: throughput, packet delay, jitter and loss rate. For the same reason as stated in Section 2, random data traffic patterns were configured, as well as a shortest path routing algorithm in the Floodlight SDN controller. To benchmark the fidelity of our test setup, a full mesh interconnection was assessed and subjected to the same tests.

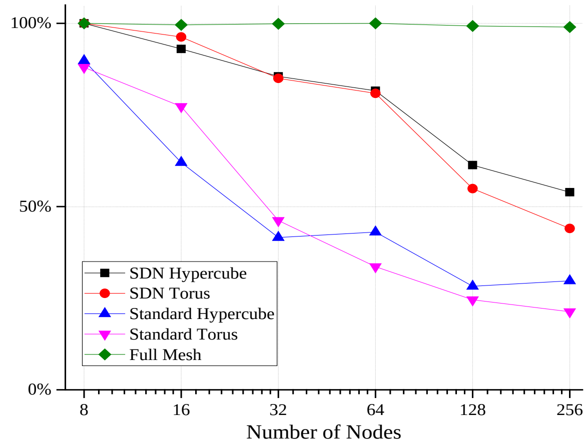


Fig. 8. Average throughput per node

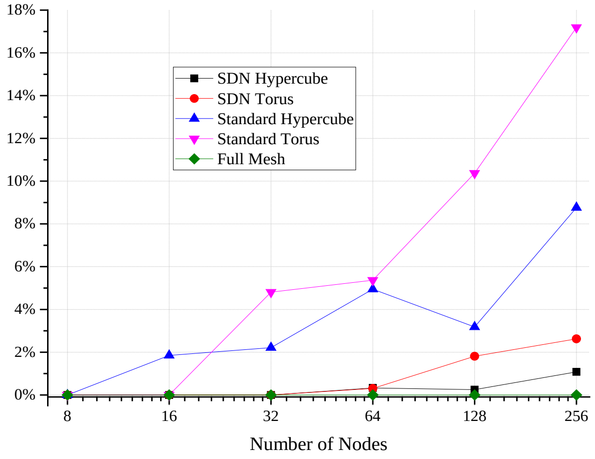


Fig. 10. Average loss rate per flow

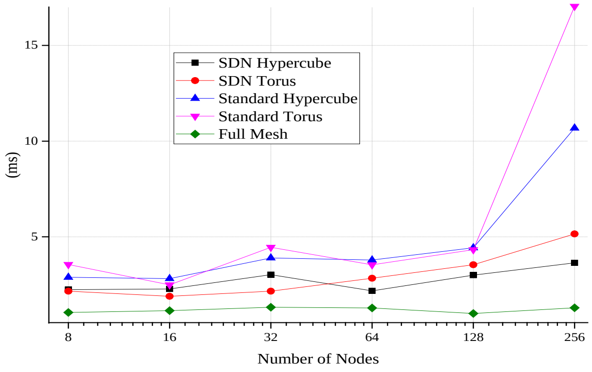


Fig. 9. Average latency per flow

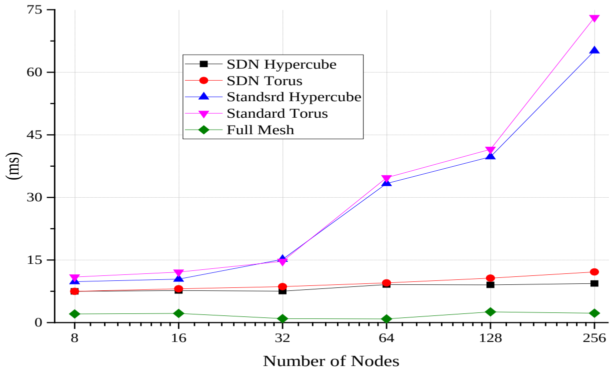


Fig. 11. Average jitter per flow

Normal traffic operation is presumed, as well as a constant injection rate (simple TCP transport) per application running on each server. Since network contention grows, the average throughput for each connection decrease when the networks expand in size including more switches and hosts.

In Figure 8, we observe the measured throughput is higher by roughly 45% for the SDN test cases with 256 nodes torus and hypercube. A 13ms decrease of packet delay, for SDN torus of the same size is noticeable in Figure 11. Loss rate is also reduced considerably for both topologies with at least 7 percentage points, as shown in Figure 9. Connection stability is improved with SDN technology by a reduction of jitter with 50ms, measurement that applies to networks of 256 switches; see Figure 10.

We plot in Figure 12 the comparison of the average throughput normalized by the theoretical link bandwidth capacity in the Jellyfish scenario. A two-fold increase in performance is observed for the SDN setup with 120 switches, with the difference slowly decreasing as the network scaled. The average packet delay, as seen in Figure 13, was considerably lower in the SDN scenario, with a small dependence on the number of switches employed and presenting less than 1/6th of the delay measured with STP for networks with more than 150 switches. Network jitter and loss rate were also more favorable in the SDN scenario: in Figure 14, we observed a reduction in jitter varying from 7% to 33%, and Figure 15 shows that the packet loss between the two systems remain within a 2% difference range independently of the number of switches.

VI. CONCLUSION AND FUTURE WORK

The first part of our paper focused on the rich and diversified design space of Data Center topologies and the differences between them. We can infer from our simulation results and the mathematical evaluation that, even though the cost indications and performance characteristics are similar for all studied topologies with under 64 nodes, clear superior operations are displayed by the hypercube followed by the torus on the downside of wiring complexity and scalability cost.

Our SDN versus STP emulation results demonstrate that the SDN deployments based on the studied topologies torus, hypercube and jellyfish, considerably outperform the STP implementation in throughput, latency, jitter and loss rate. No larger networks were tested due to the limitations of our emulation environment indicated by the degradation of the full mesh performance. Such results are representative for small to medium sized Data Centers however, much larger scales can be achieved with a distributed control plane solution [18][19], whereas the network setups discussed could represent individual clusters among many.

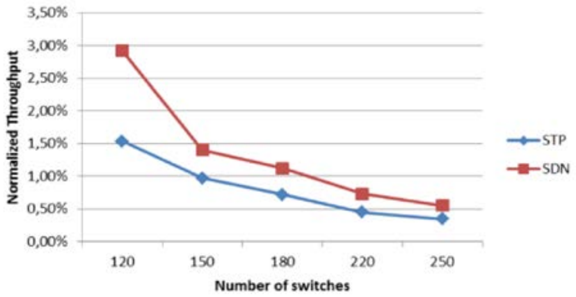


Fig. 12. Average throughput per node (Jellyfish)

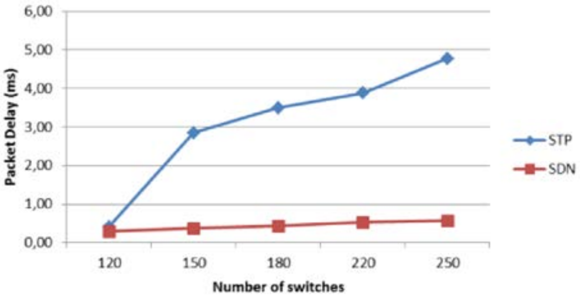


Fig. 13. Average delay per flow (Jellyfish)

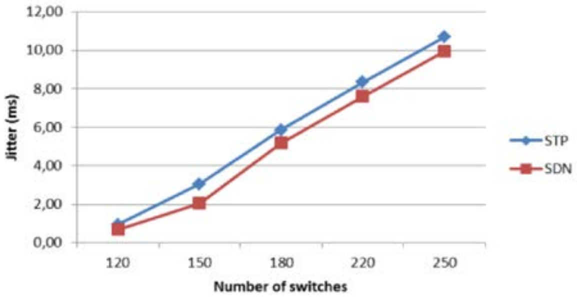


Fig. 14. Average jitter per flow (Jellyfish)

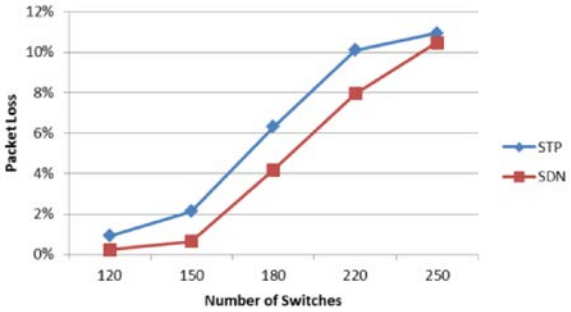


Fig. 15. Average loss rate per flow (Jellyfish)

In addition to our previously presented work [11][12], concerning the performance gains achieved with SDN in highly interconnected network topologies, these results strengthen the arguments referring to how SDN can be used to leverage the multiple redundant paths in a network. Furthermore, we believe that a combination of SDN with MultiPath Transmission Control Protocol (MPTCP) can lead to an even more efficient network infrastructure utilization.

ACKNOWLEDGEMENTS

This work has been funded by the EU FP7 Marie Curie project ABACUS and by CNPq, *Conselho Nacional de Desenvolvimento Científico e Tecnológico* – Brazil.

REFERENCES

[1] Cisco Global Cloud Index: Forecast and Methodology, 2015–2020, White Paper, 2016.

[2] M. Al-Fares, et al., “A Scalable, Commodity Data Center Network Architecture,” in Proceedings of the ACM SIGCOMM conference on Data communication (ACM), pp. 63-74, New York, 2008.

[3] G. L. Vassoler, et al. “Twin Datacenter Interconnection Topology, " in IEEE Micro, Vol.34, Issue 5, pp.8-17, 2014.

[4] H. Wu, et al. "MDCube: a high performance network structure for modular data center interconnection, " In Proc. of ACM CoNEXT, 2009.

[5] A. Greenberg et al. ,“VL2: A scalable and flexible data center network, ” in Proc. of the ACM SIGCOMM conference on Datacommunication (ACM), pp.51-62, New York, 2009.

[6] J. Chen, P. Gillard, C. Li, “Performance Evaluation of Three Network-on-Chip Architectures (Invited),” in First IEEE International Conference on Communications in China: Communications QoS and Reliability (CQR), pp. 91-96, 2012.

[7] J. K. Dennis Abts, "High Performance Datacenter Networks", by Morgan & Claypool, 2011.

[8] A. Singla et al., "Jellyfish: networking data centers randomly" in Proc. Of the 9th USENIX conference on Networked Systems Design and Implementation, pp. 17-17, California, 2012.

[9] B. Andrus, J. J. V. Olmos and I. T. Monroy, ”Performance Evaluation of Two Highly Interconnected Data Center Networks”, in Proc. International Conference on Transparent Optical Networks, Budapest, 2015.

[10] Network Simulator 3, found at: https://www.nsnam.org/.

[11] B. Andrus, J. J. V. Olmos, V. Mehmeri and I. T. Monroy „SDN data center performance evaluation of torus and hypercube interconnecting schemes, “ in Advances in Wireless and Optical Communications (RTUWO), 2015.

[12] V. Mehmeri, J. J. V. Olmos, and I. T. Monroy “Capacity Extension of Software Defined Data Center Networks With Jellyfish Topology, ” in Asia Communications and Photonics Conference (ACP), 2015.

[13] A. Singla et al.„ "Jellyfish: networking data centers randomly" in Proceedings Of the 9th USENIX conference on Networked Systems Design and Implementation, California, pp. 17-17, 2012.

[14] W. J. Dally, “Performance Analysis Of K-Ary N-Cube Interconnection Networks,” in IEEE Transactions On Computers, vol. 39, no. 6, pp. 775-785, 1990.

[15] A. S. Muhammed Mudawwar, "The k-ary n-cube Network and its Dual: a Comparative Study," in IASTED International Conference on Parallel and Distributed Computing and Systems, 2001.

[16] K. Jayaswal, “Administering Data Centers”, Wiley Publishing, 2005.

[17] T. Koponen et al., “Onix: A Distributed Control Platform for Large-scale Production Networks”, in Proc. of the 9th USENIX conference on Operating systems design and implementation, California, Article No. 1-6, 2010.

[18] A. Tootoonchian and Y. Ganjali, “HyperFlow: a Distributed Control Plane for OpenFlow”, in Proc. of the Internet Network Management Conference on Research on Enterprise Networking, California, pp.3-3, 2010.

[19] S.H Yeganeh, et al., "On scalability of software-defined networking", in IEEE Communications Magazine (Institute of Electrical and Electronics Engineers, New York), Vol. 51, Issue 2, 2013.

Simulation Study of Persistent Relay CSMA with Random Assigning of Initial Contention Window Values

Katsumi Sakakibara Naoya Yoda Kento Takabayashi
Department of Information and Communication Engineering,
Okayama Prefectural University
Soja, Japan
Email: {sakaki, cd29046m, kent.hf}@c.oka-pu.ac.jp

***Abstract*—Based on IEEE 802.11 Distributed Coordination Function (DCF), Persistent Relay Carrier Sense Multiple Access (PRCSMA) was proposed for cooperative transmission with two or more relay nodes. We propose random assigning of the initial Contention Window (CW) value of each relay node at the beginning of cooperation phase in PRCSMA. Each relay node independently and randomly selects its initial CW value among a predefined set of integers, while in the original PRCSMA, a relay node fixes its CW value to a common given integer. Numerical results obtained from computer simulation reveal that the proposed protocol can improve the performance of the original PRCSMA. The proposed protocol makes it possible to reduce the possibility of frame collisions among relay nodes and successfully reduce the duration of cooperation. Also, the results demonstrate that the binary exponential backoff algorithm degrades the performance of PRCSMA.**

Keywords–persistent relay CSMA; wireless LAN; simulation; contention window.

I. Introduction

In order to compensate poor channel quality and improve communication reliability, cooperative communications with relay nodes have been recognized as one of effective and promising techniques in wireless/mobile communication systems. Relay standards are on the way to successful implementation in Long Term Evolution (LTE)-Advanced by the Third Generation Partnership Project (3GPP) and 802.16m by IEEE [1] [2]. Relay techniques have been enthusiastically investigated from the viewpoint of the physical (PHY) and data-link layers [2] [3]. From the viewpoint of PHY layer, Multiple-Input and Multiple-Output (MIMO) and diversity techniques are known to be effective. In the data-link layer perspective, a number of Cooperative Automatic Repeat reQuest (C-ARQ) protocols have been proposed and analyzed. Particularly, the design of Medium Access Control (MAC) protocols employed between relay nodes and the destination node influences the performance, when two or more relay nodes collaborate on an identical channel.

Some MAC protocols for C-ARQ systems have been proposed recently. Morillo and Garcia-Vidal [4] proposed a C-ARQ scheme with an integrated frame combiner. They analyzed the performance with round-robin cooperation among relay nodes and with Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Alonso-Zarate et al. [5] [6] proposed Persistent Relay CSMA (PRCSMA), which elaborately incorporates well-known IEEE 802.11 Distributed Coordination Function (DCF) [7]; de facto standard for wireless LANs. When the number of relay nodes is unknown, contention-based MAC protocols are required for the data-link layer at a relay node. In [5], the performance of PRCSMA was analyzed in terms of the average duration of cooperation, based on a steady-state Markovian model proposed by Bianchi [8] For IEEE 802.11 DCF, Foh and Tantra presented an accurate three-dimensional Markovian model [9], which took into account a carry-over of backoff counter freezing after frame collision occurred. In [9], the accuracy of the new model is verified by computer simulation. In [10], the authors indicated that incorporation of Foh and Tantra's method to PRCSMA can greatly improve the performance. However, the Contention Window (CW) value of all the relay node is fixed in advance, as in the original PRCSMA [5].

In this paper, we propose random assigning of the initial CW value of each relay node at the beginning of cooperation phase in order to further improve the performance of PRCSMA. Each relay node independently and randomly selects its initial CW value among a predefined set of integers. The proposed protocol requires no information of the number of competing relay nodes. Random assigning of the initial CW value makes it possible to reduce the possibility of frame collisions among relay nodes before successful frame transmission which completes cooperative retransmissions. The performance of the proposed protocol is verified by computer simulation.

The rest of the present paper is organized as follows: Section II presents a system model with relay nodes. PRCSMA is briefly reviewed in Section III. In Section IV, the proposed protocol of random assigning of the initial CW value is described. Numerical results obtained by means of computer simulations are presented in Section V. Finally, Section VI concludes the present paper.

II. System Model

Consider a wireless network consisting of a pair of source node S and destination node D with N relay nodes; $\mathrm{R}_1, \mathrm{R}_2, \ldots,$ R_N, as shown in Figure 1. All channels are half-duplex, so that a node can not transmit and receive simultaneously. All nodes are located within their transmission range. Hence, each node can overhear ongoing transmission originating from other nodes. We assume that a node possesses no information on the number of relay nodes. Let $\varepsilon_{\mathrm{SD}}$, $\varepsilon_{\mathrm{SR}_n}$, and $\varepsilon_{\mathrm{R}_n\mathrm{D}}$ be the frame error rates on channels between source node S and destination node D, between source node S and relay node R_n, and between relay node R_n and destination node D, respectively, for $n = 1, 2, \ldots, N$. If frame transmission from source node S resulted in erroneous reception at destination node D and if one or more relay nodes succeeded in error-free reception

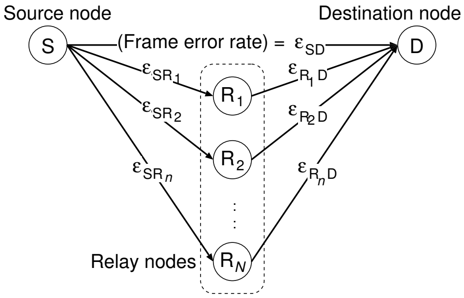


Figure 1. System model with N relay nodes.

of the frame, then such relay nodes can collaboratively serve as supporters for frame retransmission. For effective use of cooperative communications, we generally assume that $\varepsilon_{\mathrm{SD}} > \varepsilon_{\mathrm{R}_n\mathrm{D}}$. The duration in which relay nodes collaborate frame retransmissions is referred to as a *cooperation phase* [5]. Note that every frame is assumed to include an appropriate header and an ideal Frame Check Sequence (FCS) for error/collision detection, in addition to the payload. Note that the term "ideal" implies that the probability of undetected errors can be neglected.

III. PRCSMA

PRCSMA [5] [6] is a MAC protocol which elaborately resolves frame collisions among transmission from relay nodes to destination node D, based on IEEE 802.11 DCF [7]. Similarly to IEEE 802.11 DCF, each relay node in PRCSMA inserts random backoff delay before every frame transmission in a distributed manner according to its own current value of the CW. More precisely, if CW = w, then the initial value of the backoff counter is set to an integer randomly taken from the range $[0,\ w-1]$.

The operation of PRCSMA is summarized as follows. The detailed description can be found in [5]. After erroneous reception of a DATA frame transmitted by source node S, destination node D broadcasts a Call For Cooperation (CFC) frame following the Short Inter-Frame Space (SIFS). If one or more relay nodes correctly receive both the DATA frame and the CFC frame, then the cooperation phase is invoked. A relay node which joins in the cooperation phase is referred to as an *active relay node*. Active relay nodes simultaneously start the DCF operation, after the reception of the CFC frame followed by the Distributed Inter-Frame Space (DIFS). It is regulated that DIFS is longer than SIFS in order to guarantee prior transmissions of control frames to those of data frames [7]. In addition, an idle period specified by ACKtimeout after DATA frame transmission notifies nodes of transmission failure. Then, a relay node involved in collision carries out another retransmission procedure with random backoff interval. When destination node D correctly receives a DATA frame from one of the active relay nodes, it broadcasts an ACK frame to announce not only correct reception of the DATA frame to source node S but also completion of the cooperation phase to all the nodes.

In the original PRCSMA [5], the decrement of backoff counter at each relay node follows the method considered in [8]. In [10], the authors showed that by adopting the method in [9], the duration of the cooperation phase can be greatly decreased. Therefore, we consider the method in [9].

An illustrative operational example with three active relay nodes, R_1, R_2 and R_3, is shown in Figure 2. Active relay nodes R_1 and R_2 independently set their backoff counter to three and active relay node R_3 to four after reception of CFC frame from destination node D, which follows an erroneous reception of DATA frame (0). In Figure 2, a short thick down arrow marks the start of backoff interval. The first DATA frame transmissions from active relay nodes R_1 and R_2, named as DATA frames (1-1) and (2-1), respectively, result in collision. In this period of frame collision, another active relay node R_3 freezes the decrement of backoff counter. The two colliding active nodes R_1 and R_2 recognize their frame transmission failure after ACKtimeout. They randomly and independently select their next backoff interval, so that R_1 sets its backoff interval to two and R_2, to zero. Complying with the method of Foh and Tantra [9], another active relay node R_3 carries over its backoff counter whose value is one. Then, only the active relay node R_2 retransmits DATA frame (2-2). Assume that destination node D receives DATA frame (2-2) erroneously, so that the cooperation phase continues. Finally, DATA frame (1-2) is received with no errors by destination node D. Then, ACK frame transmission from destination node D notifies other nodes of completion of the cooperation phase.

IV. Random Assigning of Initial CW Values

In PRCSMA, no specific backoff algorithm such as the binary exponential backoff (BEB) algorithm is prescribed in updating the CW values of relay nodes involved in frame collisions. For the sake of mathematical tractability, Alonso-Zarate et al. [5] and Predojev et al. [6] analyzed the performance of PRCSMA with constant CW values, that is, a relay node fixes the CW value to the initial CW value, which is regulated to be equal among all the relay nodes all the time, even after frame collisions. It is clear that small CW value may increase the probability of frame collision and that large CW value may insert a large number of unnecessary idle slots, both of which may enlarge the duration of cooperation phase. From the assumption that the number of relay nodes N is unknown to all nodes, neither adaptive nor optimization techniques with respect to the CW value based on N can be applied.

In order to mitigate undesired extension of cooperation phase, we propose a random assigning of the initial CW values to each active relay node at the beginning of the cooperation phase. Let us denote the minimum and maximum CW values by $\mathrm{CW}_{\min}$ and $\mathrm{CW}_{\max}$, respectively. Here, we define a set of D possible initial CW values;

$$\mathcal{W} = \{W_0,\ W_1,\ \ldots,\ W_{D-1}\}, \qquad (1)$$

where

$$W_i = \min[2^i \mathrm{CW}_{\min},\ \mathrm{CW}_{\max}] \qquad (2)$$

for $i = 0, 1, \ldots, D-1$. In the proposed protocol, each active relay node independently and randomly selects its initial CW value among $\mathcal{W}$. For example, we have

$$\mathcal{W} = \{32,\ 64,\ 128,\ 256,\ 512,\ 1024,\ 1024\} \qquad (3)$$

for $\mathrm{CW}_{\min} = 32$, $\mathrm{CW}_{\max} = 1024$ and $D = 7$. An active relay node randomly selects one integer from $\mathcal{W}$. Since $D = 7$ and $\mathrm{CW}_{\max} = 1024$ is doubly included in $\mathcal{W}$, as in (3), each integer is selected with probability 1/7 except for 1024, whose probability is 2/7. As another choice for $\mathcal{W}$ in (3), we

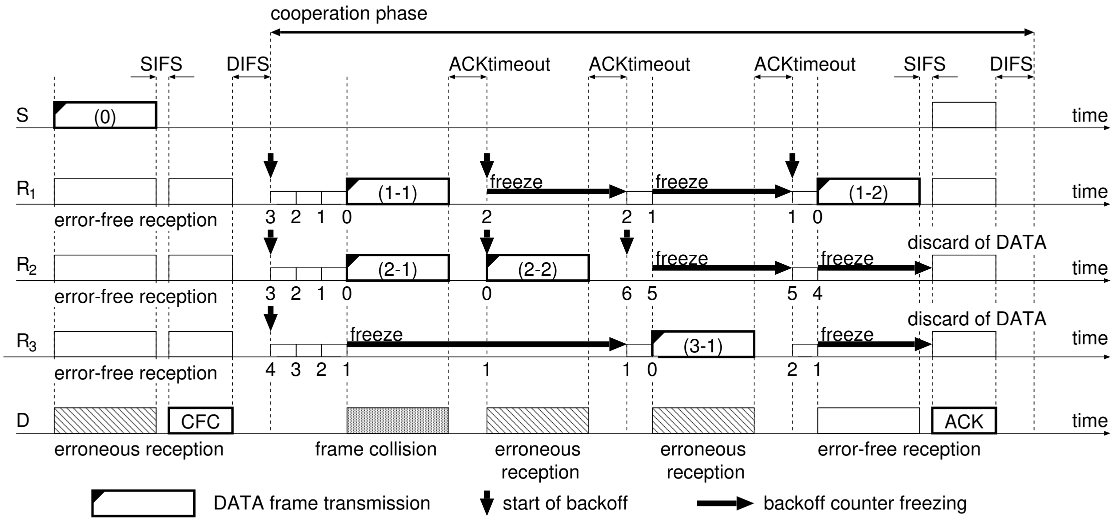


Figure 2. Illustrative example of PRCSMA with decrement of backoff counter according to Foh & Tantra's method [9].

can set $\mathcal{W} = \{32, 64, 128, 256, 512, 1024\}$ with $D = 6$ and probability 1/6 for each value. However, in the following numerical results, we permit unbalanced probabilities in order to enable us to compare the results for identical values of D.

Note here that for $D = 1$, the proposed protocol is degenerated into the original PRCSMA, since the initial CW value of all the relay node is CW_{min}.

V. Numerical Results

We evaluate the performance of the proposed protocol; random assigning of the initial CW values described in Section IV, in terms of the average duration of cooperation phase by means of exhaustive computer simulation. Comparisons with the original PRCSMA, in which each relay node fixes its CW values to the given integer all the time, are presented. We examine two cases for both the proposed protocol and the original PRCSMA. In the first case, each relay node keeps the assigned initial CW value after frame collision; without BEB, while in the second case, the CW value is doubled after frame collision until it reaches to CW_{max} in a similar manner to IEEE 802.11 DCF; with BEB. The simulation program is written in C language and the results are obtained by averaging 10^5 trials of cooperation phases. Each trail starts with N active relay nodes, which implies that all the relay nodes correctly receive both DATA frame from source node S and CFC frame from destination node D. The values of parameters used in simulations are tabulated in Table I, which are basically taken from IEEE 802.11a standard [7]. The values of CW_{min} are taken by referring to IEEE 802.11e standard. Channels between relay node R_n and destination node D are assumed error-free; $\varepsilon_{\text{R}_n\text{D}} = 0$ for any $n = 1, 2, \ldots, N$. Hence, frame transmission succeeds if it experiences no other simultaneous frame transmissions.

A. Average Duration of Cooperation Phase

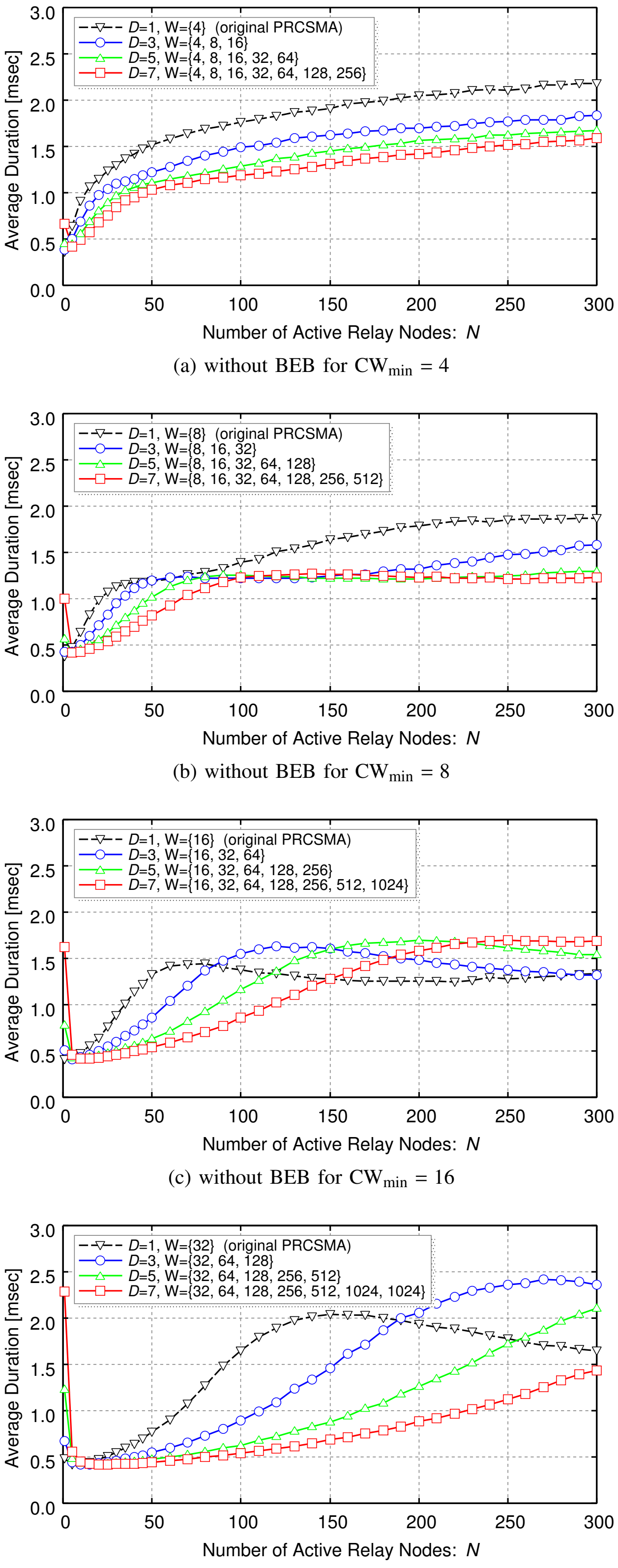
The average duration of cooperation phase of the proposed protocol and the original PRCSMA ($D = 1$) is shown for $\text{CW}_{\text{min}} = 4, 8, 16, 32$ in Figure 3 and Figure 4. In Figure 3, no BEB algorithm is employed, so that each relay node holds the assigned initial CW value after frame collision. In Figure 4, each relay node doubles its CW value unless it is greater than CW_{max}. Shorter duration of cooperation phase is preferred, since nodes can move to the next data transfer rapidly.

TABLE I. Parameters Used in Simulations.

data rate	54	[Mbps]
control frame rate	6	[Mbps]
slot duration	9	[μsec]
SIFS duration	16	[μsec]
DIFS duration	34	[μsec]
ACKtimeout	34	[μsec]
round-trip time	0	[μsec]
PHY header length	20	[μsec]
MAC header length	34	[byte]
ACK length	14	[byte]
DATA payload length	1500	[byte]
CW_{min}	4,8,16,32	
CW_{max}	1024	
D (size of $\mathcal{W}$)	1,3,5,7	
frame error rate $\varepsilon_{\text{R}_n\text{D}}$	0	(error-free)

First, let us roughly compare the performance between the proposed protocol and the original PRCSMA. If the number of active relay nodes N is small, then the proposed protocol for large D is preferred. This tendency is outstanding for wider range of N, when CW_{min} is smaller. On the other hand, if N is large, the original PRCSMA exhibits small average duration of cooperation phase. As a whole, we can observe from Figure 3 and Figure 4 that the proposed protocol for $\text{CW}_{\text{min}} = 8$ and $D = 7$ with no BEB algorithm indicates best average duration of cooperation phase, which is stable for wide range of the number of active relay nodes.

Next, compare the performance with and without BEB algorithm. An incorporation of the BEB algorithm generally enlarge the average duration of cooperation phase. An effectiveness of the BEB algorithm has been widely known and analyzed. In fact, the BEB algorithm is able to reduce the probability of frame collision and to improve the steady-state performance. However, it is revealed from Figure 3 and Figure 4 that the BEB algorithm may defer the occurrence of the first successful frame transmission, in particular, in dense networks. The results give us an insight that in the transient state, the CW values should be kept constant until some frames



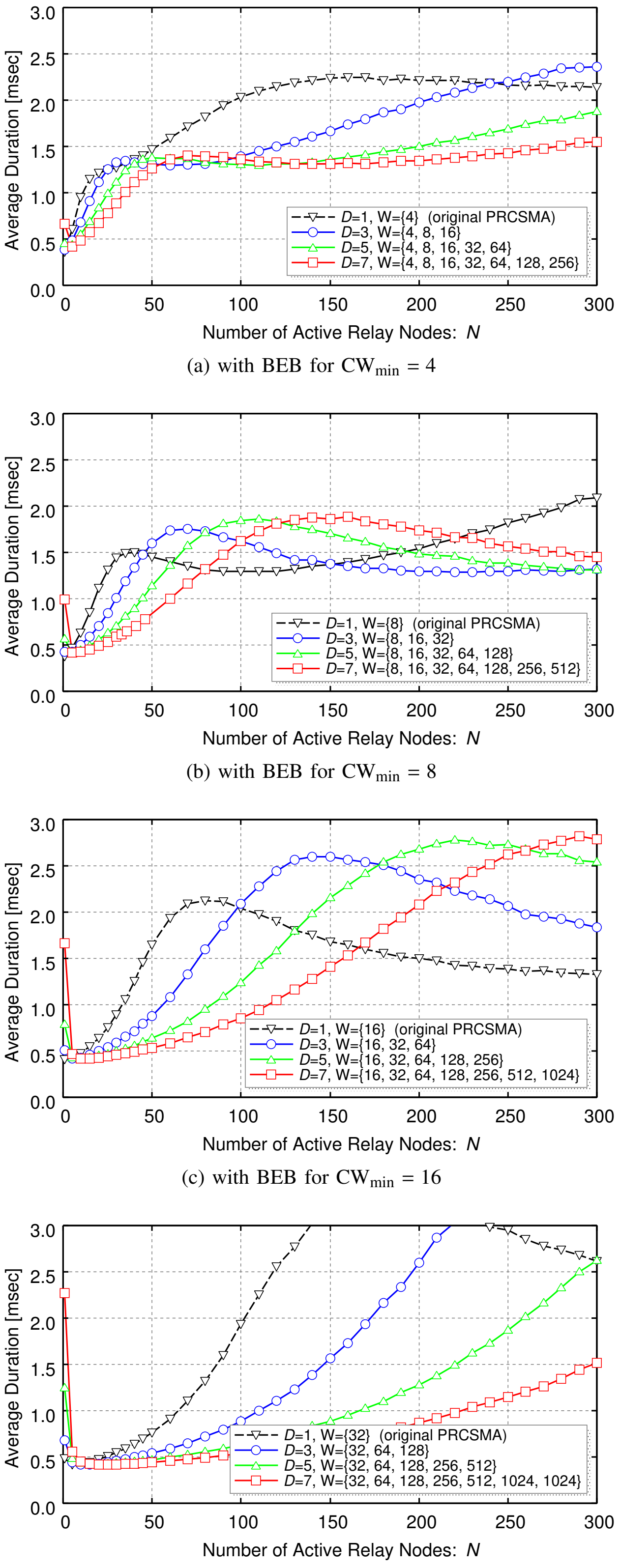
(a) without BEB for CW$_{\min}$ = 4

(b) without BEB for CW$_{\min}$ = 8

(c) without BEB for CW$_{\min}$ = 16

(d) without BEB for CW$_{\min}$ = 32

Figure 3. Average duration of cooperation phase without BEB algorithm for CW$_{\min}$ = 4, 8, 16, 32 and D = 1, 3, 5, 7.



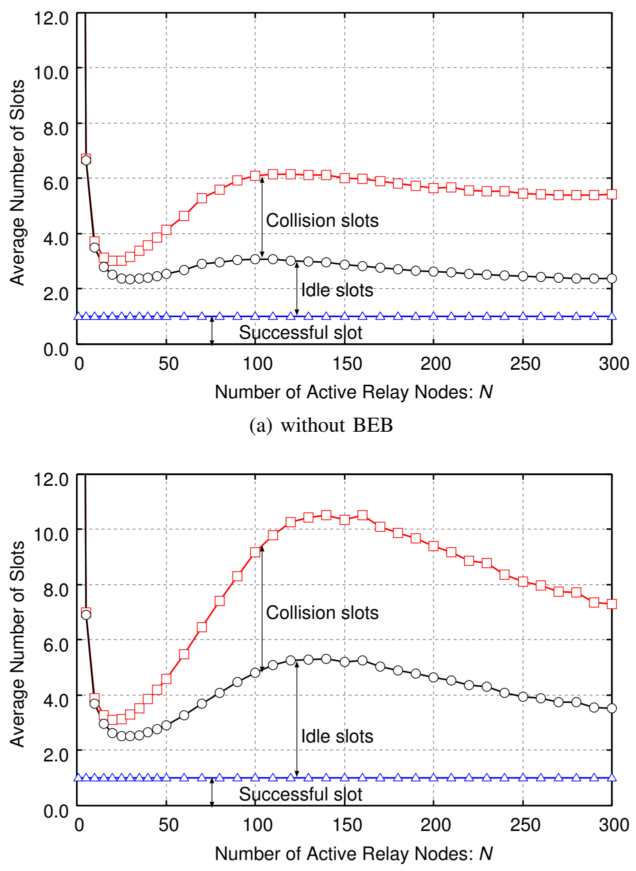
(a) with BEB for CW$_{\min}$ = 4

(b) with BEB for CW$_{\min}$ = 8

(c) with BEB for CW$_{\min}$ = 16

(d) with BEB for CW$_{\min}$ = 32

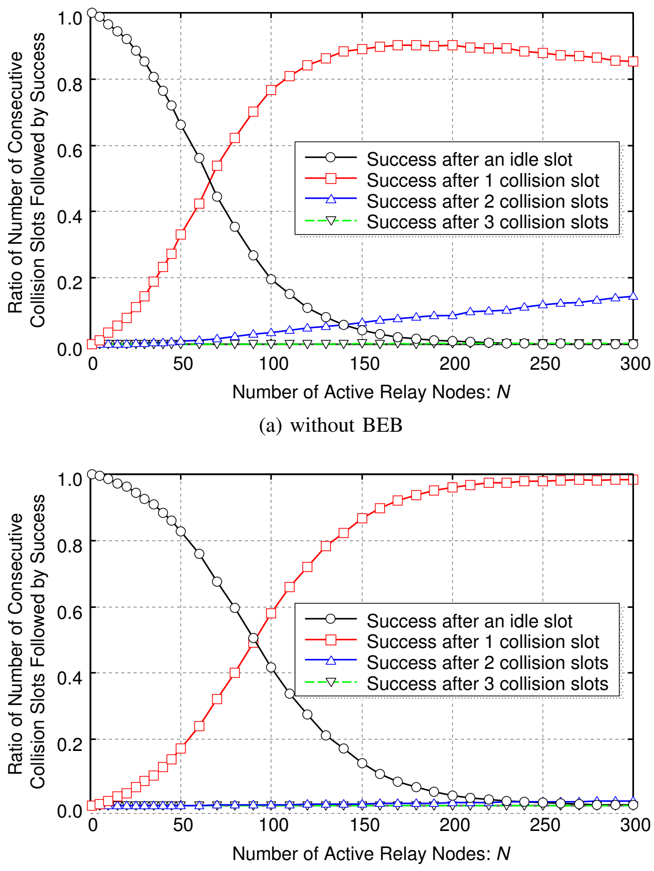
Figure 4. Average duration of cooperation phase with BEB algorithm for CW$_{\min}$ = 4, 8, 16, 32 and D = 1, 3, 5, 7.



(a) without BEB

(b) with BEB

Figure 5. Distribution of average number of slots in cooperation phase for $CW_{min} = 8$ and $D = 7$.



(a) without BEB

(b) with BEB

Figure 6. Ratio of the number of consecutive collisions followed by successful DATA frame transmission of the proposed protocol and the original PRCSMA for $CW_{min} = 8$ and $D = 7$.

succeed in transmission, and then the BEB algorithm should be invoked.

B. Slot Distribution

In order to reveal the reason why the adoption of the BEB algorithm may bring about longer unnecessary time before the first success of frame transmission, we examine the average number of virtual slots in a cooperation phase. Since channel errors between relay nodes and destination node are ignored, virtual slots can be classified into idle, collision, and successful slots, where every cooperation phase ends with a unique successful slot.

We examine the distribution of the number of slots in a cooperation phase in average. The numerical results are shown in Figure 5 for the case of $CW_{min} = 8$ and $D = 7$. The results with no BEB algorithm are given in Figure 5(a) and those with the BEB algorithm, in Figure 5(b). Comparing two graphs in Figure 5, we can find that the number of collision slots and idle slots in Figure 5(b) increases for $N > 50$. For $CW_{min} = 8$ and $D = 7$, $N/7$ relay nodes start a cooperation phase with $CW = 8$. If N is less than 50, about seven relay nodes start with $CW = 8$ and other relay nodes, with $CW \in \{16, \ldots, 1024\}$. Therefore, the probability of frame collision may be small, since the most possible collision among relay nodes with $CW = 8$ may be rare. For $N > 50$, the probability of frame collision can be expected to increase, which causes a necessity of frame retransmissions. Recall here that according to Foh and Tantra's method [9], only the relay nodes involved frame collision are permitted to retransmit their frame in the next time slot, if their new backoff counter is zero. It implies that the possibility of consecutive occurrence of frame collision in the time slot following frame collision can be mitigated. However, the doubling process of CW values in the BEB algorithm may decrease the possibility to randomly select zero backoff counter, compared to the case without the BEB algorithm.

C. Consecutive Frame Collisions

Next, we evaluate the number of consecutive frame collisions followed by a successful DATA frame transmission, which entails the end of the cooperation phase.

In Figure 6, the ratio of the number of consecutive frame collision slots followed by successful DATA frame transmission is shown for the proposed protocol and the original PRCSMA for $CW_{min} = 8$ and $D = 7$ with and without the BEB algorithm. The results with no BEB algorithm are given in Figure 6(a) and those with the BEB algorithm, in Figure 6(b). An illustrative description of the number of consecutive collisions followed by successful frame transmission is shown in Figure 7. It follows from Figure 6(a) that in the case of with no BEB algorithm, the ratio of successful DATA frame transmission following an isolated frame collision slot after an idle slot; red curve, increases faster than the case with the BEB algorithm for $50 < N < 150$. For $N > 150$, red curve for the case with the BEB algorithm is greater than that for the

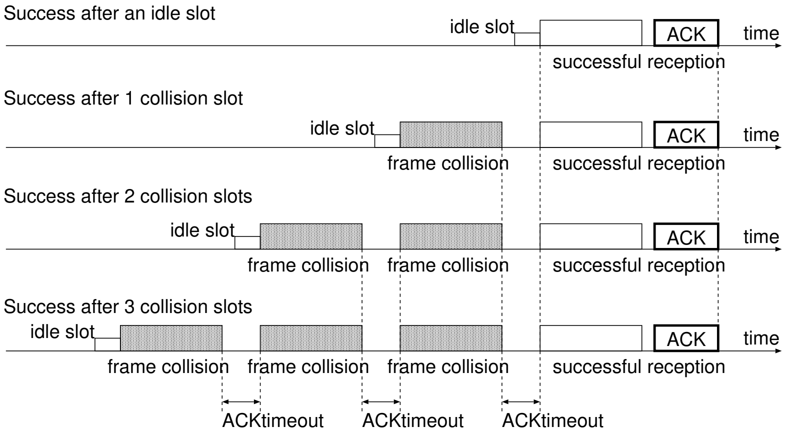


Figure 7. Description of the number of consecutive frame collisions followed by successful DATA frame transmission in Figure 6.

case without BEB algorithm. However, since the possibility of two or more consecutive frame collisions decreases in the case with the BEB algorithm, the corresponding curves; blue and black curves, are almost zero. This means that a number of idle slots are inserted in the case of BEB algorithm before successful frame transmission.

VI. Conclusion and Future Work

Random assigning of the initial CW value of each relay node at the beginning of cooperation phase has been proposed in PRCSMA. Each relay node independently and randomly selects its initial CW value among a predefined set of integers, while in the original PRCSMA, a relay node fixes its CW value to a common integer. Numerical results obtained from computer simulation have revealed that the proposed protocol can improve the performance of the original PRCSMA. The proposed protocol makes it possible to reduce the possibility of frame collisions among relay nodes before successful frame transmission which completes a cooperation phase. Also, the results demonstrate that the binary exponential backoff algorithm degrades the performance of PRCSMA.

Further work includes, for example, the extension to bidirectional communication systems and to the use of network coding. Also, the theoretical analysis through appropriate mathematical modeling of the proposed protocol should be investigated.

Acknowledgment

A part of this work was supported by Japan Society for the Promotion of Science (JSPS) under Grant-in-Aid for Scientific Research (C) (KAKENHI No. 16K06363).

References

[1] K. Loa, et al., "IMT-advanced relay standards," IEEE Commun. Mag., vol. 48, no. 8, pp. 40–48, Aug. 2010, doi: 10.1109/MCOM.2010.5534586.

[2] A. Bhamri, F. Kaltenberger, R. Knopp, and J. Hamalainen, "Smart hybrid-ARQ (SHARQ) for cooperative communication via distributed relays in LTE-advanced," Proc. IEEE Intl. Workshop on Signal Processing Advances in Wireless Commun. (SPAWC 2011), San Francisco, CA, June 2011, pp. 41–45, doi: 10.1109/SPAWC.2011.5990443.

[3] F. Gomez-Cuba, R. Asorey-Cacheda, and F. J. Gonzalez-Castano, "A survey on cooperative diversity for wireless networks," IEEE Commun. Surveys & Tutorials, vol. 14, no. 3, pp. 822–835, 3rd Qtr, 2012, doi: 10.1109/SURV.2011.082611.00047.

[4] J. Morillo and J. Garcia-Vidal, "A cooperative-ARQ protocol with frame combining," Wireless Networks, vol. 17, no. 4, pp. 937–953, May 2011, doi: 10.1007/s11276-011-0326-y.

[5] J. Alonso-Zarate, L. Alonso, and C. Verikoukis, "Performance analysis of a persistent relay carrier sensing multiple access protocol," IEEE Trans. Wireless Commun., vol. 8, no. 12, pp. 5827–5831, Dec. 2009, doi: 10.1109/TWC.2009.12.090707.

[6] T. Predojev, J. Alonso-Zarate, L. Alonso, and C. Verikoukis, "Energy efficiency analysis of a cooperative scheme for wireless local area networks," Proc. IEEE Global Commun. Conf. (GLOBECOM 2012), Anaheim, CA, Dec. 2012, pp. 3183–3186, doi: 10.1109/GLOCOM.2012.6503604.

[7] IEEE Standard 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, Piscataway, NJ, 1999.

[8] G. Bianchi, "Performance analysis of the IEEE 802.11 distribution coordination function," IEEE J. Select. Areas Commun., vol. 18, no. 3, pp. 535–547, Mar. 2000, doi: 10.1109/49.840210.

[9] C. H. Foh and J. W. Tantra, "Comments on IEEE 802.11 saturation throughput analysis with freezing of backoff counters," IEEE Commun. Lett., vol. 9, no. 2, pp. 130–132, Feb. 2005, doi: 10.1109/LCOMM.2005.02008.

[10] K. Sakakibara, T. Harada and J. Taketsugu, "Performance approximation of persistent relay CSMA with carry-over of backoff counter freezing after collision," WSEAS Trans. Commun., vol. 14, Article ID 1, pp. 1–10, 2015.

A Preliminary Study on Using Smartphones to Detect Falling Accidents

Jin-Shyan Lee and Hsuan-Han Tseng
Department of Electrical Engineering
National Taipei University of Technology (Taipei Tech.)
Taipei, Taiwan
jslee@mail.ntut.edu.tw and t102318057@ntut.edu.tw

***Abstract*—In order to improve the disadvantages of current smartphone-based fall detection systems, this paper analyzes the characteristics of triaxial accelerometer values to identify thresholds of the non-falls and falls, and proposes an improved threshold-based fall detection method, which is not only able to quickly filter out most of daily activities (including walking running, sitting down, and so on) but also to detect the direction of four types of falling events. Moreover, as soon as a falling accident is detected, the user's position could be immediately sent to the rescue center so as to get medical help.**

Keywords*—*fall detection; smartphones; triaxial accelerometers.

I. INTRODUCTION

In order to solve the fall crisis that the elderly will be faced with in daily life, there are many research scholars devoted to the research field of fall detection [1]. The fall detection methods could be mainly divided into environmental detection type and wearable detection type. Environmental detection type is mainly to place sensors in the detective areas of daily life. Shieh and Huang [2] collected the monitored images from different areas by placing numerous surveillance cameras and proposed using a pattern recognition approach to detect the elder's falling event. However, the disadvantage of this approach is that fall detection is limited to the monitored environment. Also, the privacy issue of users is a problem. In order to overcome the disadvantages of this type, many research scholars come up with a wearable detection method, in which the user wears sensors so as to provide human activity data for fall detection. Cheng and Jhan [3] used tri-axis acceleration sensor with the proposed cascade-AdaBoost-support vector machine (SVM) classifier. The algorithm could automatically determine whether to replace the AdaBoost classifier by SVM. The results are compared to those of the neural network, SVM, and the cascade-AdaBoost classifier. The experimental results show that the triaxial accelerometers around the chest and waist produce optimal results, and our proposed method has the highest accuracy rate and detection rate as well as the lowest false alarm rate.

Tong *et al.* [4] proposed a hidden Markov model (HMM)-based method to detect and predict falls using triaxial accelerations of a human body. The acceleration time series extracted from human motion processes are used to describe human motion features and train HMM so as to build a random process mathematical model. Thus, the outputs of HMM could be used to evaluate the risks to fall. The experiment results showed that fall events can be predicted 200-400 ms ahead of the occurrence of collisions, and distinguished from other daily life activities with an accuracy of 100%.

With the popularization of smartphones, the mobile phone has become an indispensable product in our daily life. Therefore, in recent years, many researchers integrate the smartphone into fall detection study. In [5], the authors demonstrated techniques to detect a fall and also automatically classify the type. Four different types of falls, left and right lateral, forward and backward falls are discussed and five machine learning classifiers are applied to a large time-series feature set to detect falls. The results showed that SVM and regularized logistic regression were able to identify a fall with 98% accuracy and classify the type of fall with 99% accuracy. In [6], the angles acquired by the electronic compass and the waveform sequence of the triaxial accelerometer on the smartphone are used to generate an ordered feature sequence and then examined in a sequential manner by their proposed cascade classifier for fall detection. The experimental results show that a fall accident detection accuracy up to 92% on the sensitivity and 99.75% on the specificity could be obtained.

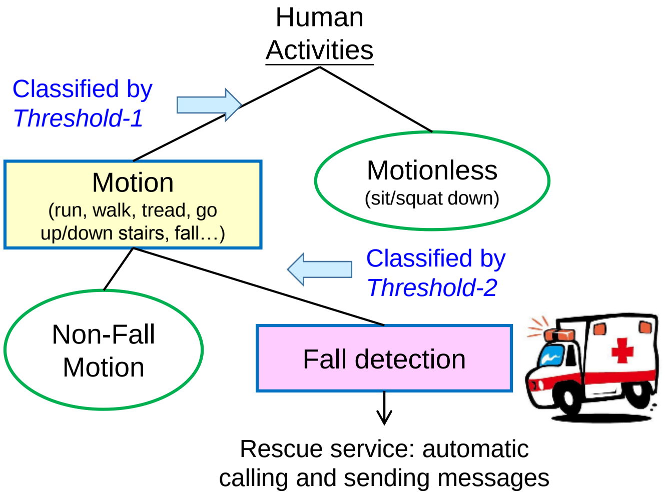


Figure 1. Proposed threshold-based fall detection scheme.

In this paper, through experimentally analyzing the characteristics of triaxial accelerometer values, a threshold-based fall detection approach has been proposed. As shown in Figure 1, the human activities are first classified into motion or motionless category by the *Threshold-1*. Then, in the motion class, including the actions of running, walking, tread, going up/down stairs, as well as falling, the *Threshold-2* is employed to recognize the fall events. After detecting a falling accident, a rescue service will be carried out by automatic calling and sending messages to the emergency center so as to immediately get medical help.

Copyright (c) IARIA, 2017. ISBN: 978-1-61208-591-3

The rest of this paper is organized as follows. Section II briefly introduces the proposed scheme. Next, preliminary experiments are described in Section III. Finally, Section IV concludes this paper.

II. PROPOSED METHODS

As shown in Figure 2, this paper adopts an Android-based smartphone, which is assumed to be carried in a front pants pocket, as the development platform for fall detection. Moreover, a sampling rate of 50 Hz is used to collect sensing data from the build-in triaxial accelerometer in a smartphone.

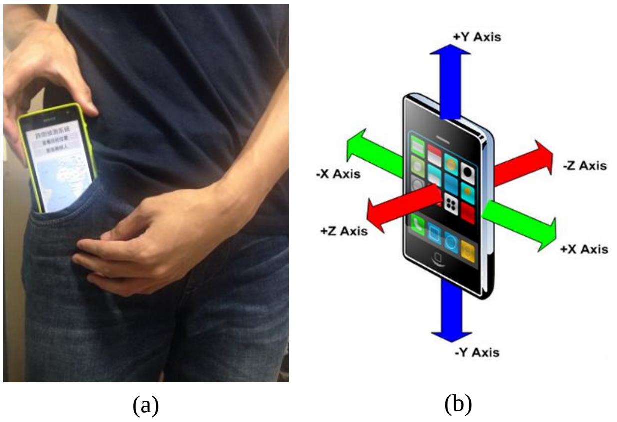


Figure 2. (a) Smartphone position in a front pants pocket. (b) Three axes of a smartphone.

A. Motion Activities Recognition

Through observing triaxial accelerometer's values, it could be found that the acceleration changes significantly when a fall occurs. Therefore, the change of acceleration intensity value can be used to determine whether a fall occurs. In this paper, the signal magnitude area (SMA) value, defined as (1), is used to determine the motion and motionless activities.

$$\mathrm{SMA}[n] = \frac{1}{N}\left(\sum_{i=n-N+1}^{n} |x[i]| + \sum_{i=n-N+1}^{n} |y[i]| + \sum_{i=n-N+1}^{n} |z[i]|\right) \tag{1}$$

where $x[n]$, $y[n]$, and $z[n]$ are the acceleration values of the three axes, respectively, at the sampling time n.

Through several practical experiments, we found as the user is engaged in a motion activity, the SMA value is much higher than that in motionless ones, such as sitting and squatting down. Figure 3 shows the SMA values for human activities, including falling down, sitting down, and squatting down. Table I shows the nine types of human activities and its corresponding SMA values (the signs + means more and – means less, respectively). Hence, the *Threshold-1* is determined as 27 $\mathrm{m/s^2}$ via experimental observations. It is noted that even though this parameter is not theoretically proved, the feasibility would be later verified via experimental results.

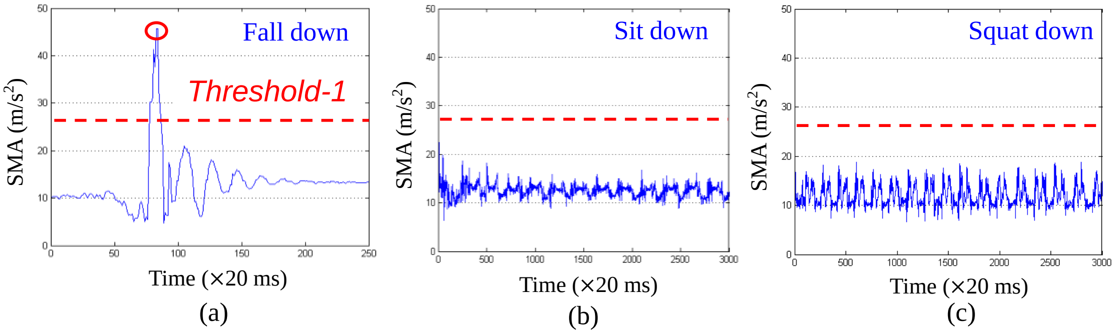


Figure 3. SMA values for human activities: (a) fall down, (b) sit down, and (c) squat down.

TABLE I
SMA VALUES FOR HUMAN ACTIVITIES

Activities	Fall down	Run	Walk	Tread	Go downstairs	Go upstairs	Sit down	Squat	Stand
SMA ($\mathrm{m/s^2}$)	30 +	30 +	30 +	30 +	30 +	30 +	25 -	25 -	25 -

Threshold = 27 (determined via experimental observations)

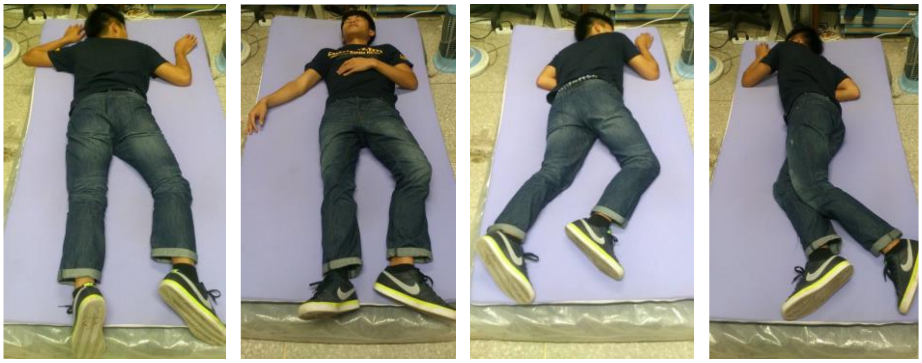
B. Falling Accidents Detection

In the proposed scheme, four types of falls are considered, including the forward, backward, left lateral, and right lateral falls, as shown in Figure 4.

In order to detect falling accidents, the mean values of acceleration of single axis would be employed. Since the smartphone is assumed to be placed in the front pants pocket, the mean values of x-axis acceleration could be used to detect lateral falls towards the right and left direction. On the other hand, the mean values of z-axis acceleration would be used to forward and backward fall detection. As an example, the mean values x-axis acceleration is defined as (2).

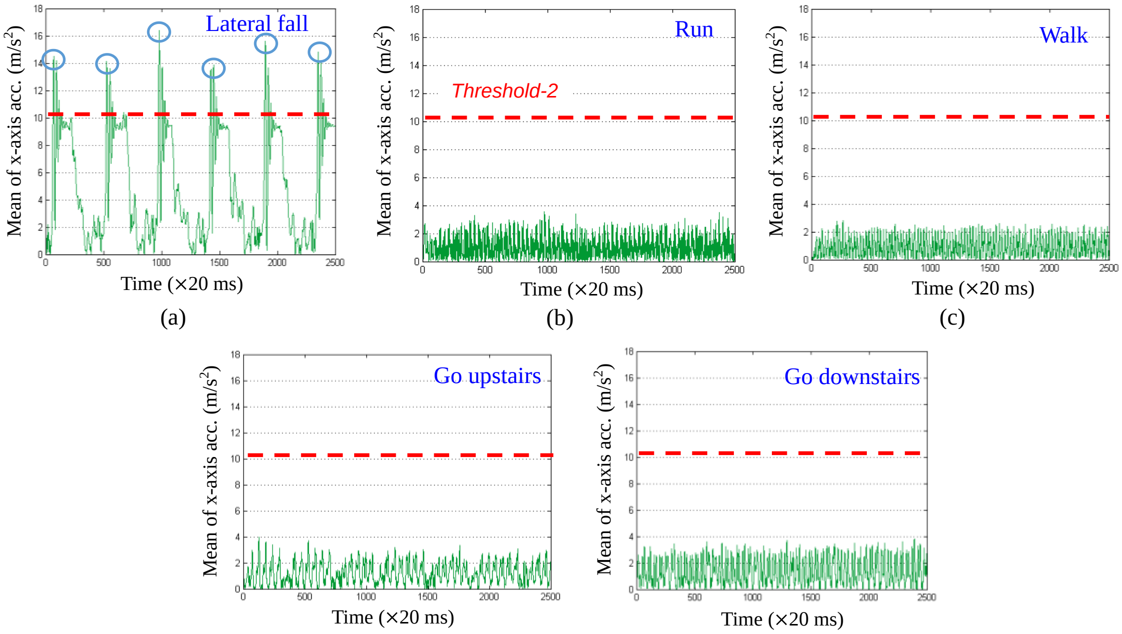
$$x_{\text{mean}}[n] = \frac{1}{N}\left(\sum_{i=n-N+1}^{n} |x[i]|\right) \tag{2}$$

Figure 5 shows the mean values of x-axis acceleration values for human activities, including the lateral fall, run, walk, go upstairs, and go downstairs. Hence, the *Threshold-2* is determined as 10.05 $\mathrm{m/s^2}$ via experimental observations. It is noted that even though this parameter is not theoretically proved, the feasibility would be later verified via experimental results.



(a) (b) (c) (d)

Figure 4. (a) Forward, (b) backward, (c) left lateral, and (d) right lateral falls.



(a) (b) (c)

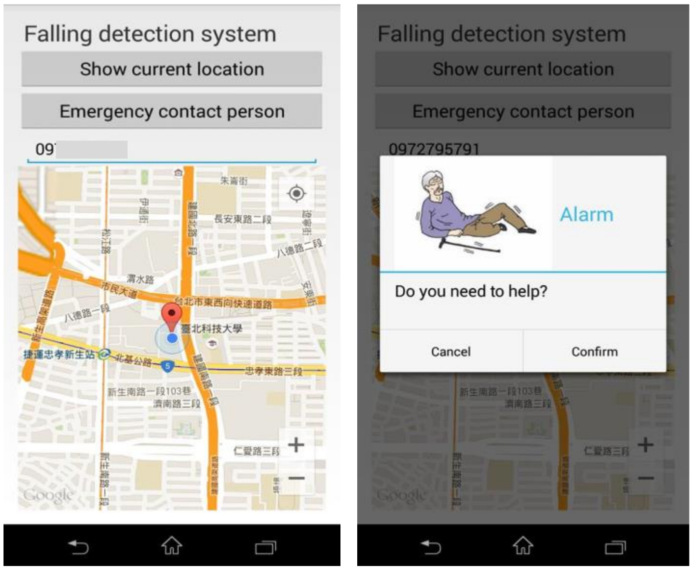
(d) (e)

Figure 5. Mean of x-axis acceleration values for human activities: (a) lateral fall, (b) run, (c) walk, (d) go upstairs, and (e) go downstairs.

III. Implementation and Experiments

A. System Implementation

The proposed fall detection approach has been implemented as an APP with the user interface as shown in Figure 6. In addition, the message flows among the user, fall detection APP, and emergency center are shown in Figure 7. Moreover, an Android-based smartphone with the specification shown in Table II [7] is applied to conduct the experiments.



Falling detection system

Show current location

Emergency contact person

09[redacted]

Falling detection system

Show current location

Emergency contact person

0972795791

Alarm

Do you need to help?

Cancel

Confirm

Figure 6. The developed user interface of the fall detection system.

TABLE II
Specifications of the Smartphone [7]

Type	Sony Xperia TX
OS	Android 4.3
Size	4.6 inch
Resolution	1280 x 720 pixels
CPU	Qualcomm S4 MSM8260A - 1.5GHz
RAM	1GB
ROM	16GB
Communication	3G、GPS、Bluetooth、Wi-Fi
Sensor	Tri-axial accelerometer (± 20 G)

B. Experimental Results

To conduct the evaluation process, nine different kinds of activities including a fall down event, running, walking, sitting down, going upstairs, going downstairs, tread, standing up, and squatting have been evaluated, each with 50 tests. In order to assess the testing effect, accuracy rate (AR), detection rate (DR), and false alarm rate (FAR) are expressed as (3), (4), and (5), respectively.

$$\mathrm{AR} = (\mathrm{TP} + \mathrm{TN})/(p + q) \tag{3}$$

$$\mathrm{DR} = \mathrm{TP}/p \tag{4}$$

$$\mathrm{FAR} = \mathrm{FP}/q \tag{5}$$

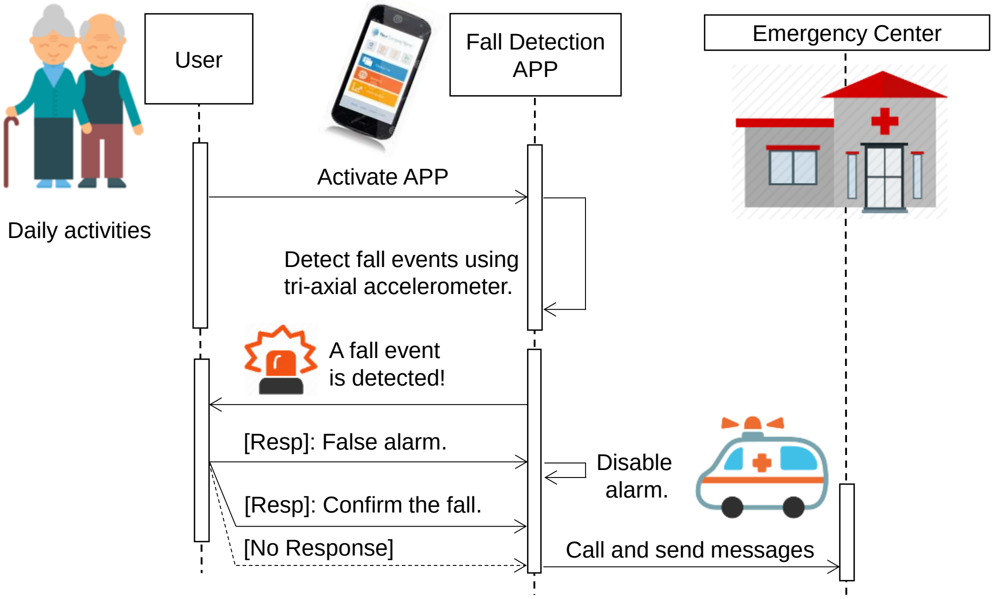


Figure 7. Message flow among the user, fall detection APP, and emergency center.

ISBN: 978-1-61208-591-3

TABLE III
EXPERIMENT RESULTS OF DETECTING FOUR FALL DIRECTIONS

Our design	Run	Walk	Sit down	Go upstairs	Go downstairs	Tread	Stand up	Squat	Fall down (4 types)
Test samples	50	50	50	50	50	50	50	50	100
TP	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	96
FP	0	0	1	0	0	0	0	0	N/A
TN	50	50	49	50	50	50	50	50	N/A
AR (accuracy rate)	99%								
DR (detection rate)	96%								
FAR (false alarm rate)	0.25%								
Computation time	17.8ms								

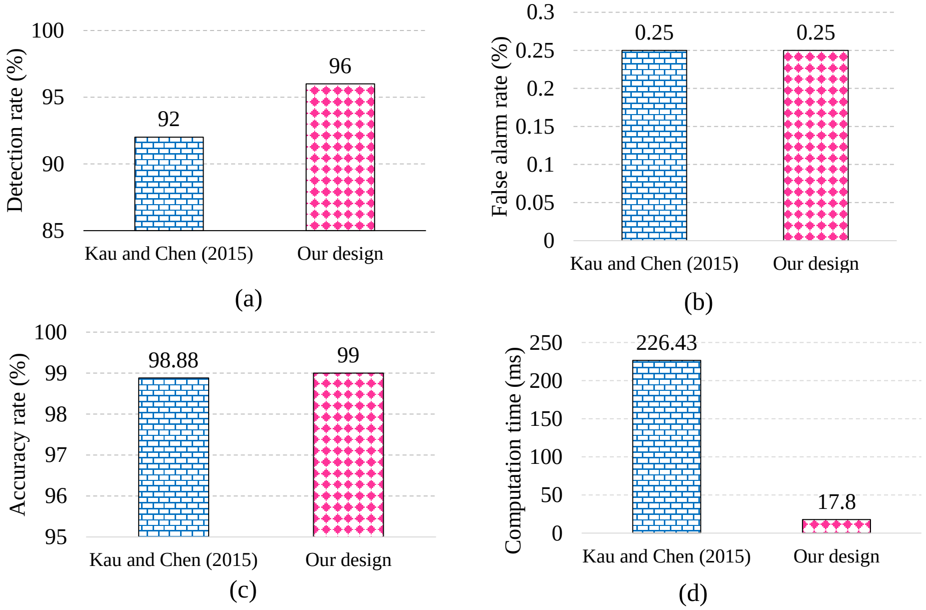


Figure 8. Comparison of (a) detection rate, (b) false alarm rate, (c) accuracy rate, and (d) computation time.

wherein p and q mean the number of collections of the positive examples (falls) and negative examples (non-falls), respectively. The true positive (TP) represents the number of falls successfully detected, true negative (TN) indicates the number of non-fall examples successfully detected, and false positive (FP) shows the number of non-fall examples detected as a fall.

The experiments conducted in this paper are based on the most frequent human daily activities, which include running, walking, sitting down, going upstairs, going downstairs, tread, standing up, and squatting, as well as the four types of falls. The experimental results of detecting four fall directions are shown in Table III. The AR, DR, and FAR are 99%, 96%, and 0.25%, respectively. Moreover, the computation time is 17.8 ms. Figure 8 compares the results of the proposed approach with [6]. It is clear that the proposed method is better in terms of AR DR, and computation time.

IV. CONCLUSION AND FUTURE WORK

This paper has proposed a fall detection system for the elderly by using smartphones placed in their pants pockets. By analyzing acceleration characteristics of three axes x, y, z values, effective threshold values to distinguish daily activities and falling accident have been determined. The results show the proposed method not only has the accuracy rate about 99% but also consumes less computation time, which greatly reduces the burden of the mobile phone operation. In the current scheme, the two thresholds are fixed. Future work will attempt to develop adjustable thresholds considering different users.

ACKNOWLEDGMENT

This paper was supported by the Ministry of Science and Technology (MOST) of Taiwan under the grant 105-2221-E-027-069.

REFERENCES

[1] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," IEEE Trans. Inform. Technology in Biomedicine, vol. 10, no. 1, pp. 156-167, Jan. 2006.

[2] W. Y. Shieh and J. C. Huang, "Speedup the multi-camera video-surveillance system for elder falling detection," in Proc. IEEE. Int. Conf. Embedded Software and Systems, Zhejiang, China, May 2009, pp. 350-355.

[3] W. C. Cheng and D. M. Jhan, "Triaxial accelerometer-based fall detection method using a self-constructing cascade-AdaBoost-SVM classifier," IEEE J. Biomedical and Health Inform., vol. 17, no. 2, pp. 411-419, Mar. 2013.

[4] L. Tong, Q. Song, Y. Ge, and M. Liu, "HMM-based human fall detection and prediction method using triaxial accelerometer," IEEE. Sensors J, vol. 13, no. 5, pp. 1849-1856, May 2013.

[5] M. V. Albert, K. Kording, M. Herrmann and A. Jayaraman, "Fall classification by machine learning using mobile phones," PLoS ONE, J. Biomed. Eng. Online, vol. 7, no. 5, pp. 1-6, May 2012.

[6] L. J. Kau and C. S. Chen, "A smart phone-based pocket fall accident detection, positioning and rescue system," IEEE J. Biomedical and Health Inform., vol. 19, no. 1, pp. 44-56, Jan. 2015.

[7] Sony Xperia TX Specifications, 2012. [Online] Available: https://zh.wikipedia.org/wiki/Sony_Xperia_TX

Evaluation of Packet Preemption over C-RAN Fronthaul Networks

Ying Yan, Zifan Zhou, Sarah Ruepp, and Michael Stübert Berger
Department of Photonics Engineering
Technical University of Denmark, DTU
Kgs. Lyngby, Denmark
E-mail: yiya@fotonik.dtu.dk

***Abstract*—The Cloud Radio Access Network (C-RAN) is viewed as a new solution with benefits of reduced cost by sharing resources. This is achieved by the separation of the radio part and the radio processing part, where the transport network between them is referred to as front-haul. It is essential to meet the stringent service requirements of protocols running over the front-haul. This paper describes the C-RAN features and challenges. Furthermore, this paper verifies the packet preemption technology in the C-RAN based on both numerical analysis and simulation results.**

Keywords— time-sensitive network (TSN); C-RAN; packet preemption; preemptive queueing

I. INTRODUCTION

The stringent delay and jitter requirements become a crucial constraint for various applications in reality. When the network operators serve the multimedia streaming services, the quality of received video data is degraded if the delay and jitter requirements cannot be satisfied. When the factories operate the machine production line or the robot line over a remote control, the eventual manipulation can be mismatched with the commands if the control signal cannot be transmitted within the demanding delay requirements. When the car manufactories introduce the advanced techniques, such as Infotainment, Telematics and Advanced Driver Assistance System (ADAS) in the vehicle, the expected convenience and safety cannot be ensured by using traditional electronic components and systems without a suitable end-to-end delay guarantee. This paper discusses the improvement on a network switch in order to differentiate and handle critical traffic with low delay.

Packet preemption has been developed by the IEEE 802.1 Time Sensitive Networking (TSN) work group [1]. In TSN, the control traffic can be scheduled and transferred by using a time-triggered method. There is a specific time window reserved for the arrival of a control packet. In allocating the time window to be as close to the arrival time as possible, a preemptive based priority scheduling is supported. The interfered traffic becomes preemptive and is thus allowed to be interrupted during transmission. Therefore, a minimum end-to-end delay is ensured for the control traffic.

In this paper, we integrate the TSN technology packet preemption for the Cloud based Radio Access Networks (C-RAN). In C-RAN, a mobile operator’s radio equipment and the controller are separated geographically and the connection link between them is essential to meet the stringent service requirements. We contribute to verify the TSN benefits for the C-RAN based on both numerical analysis and simulation results.

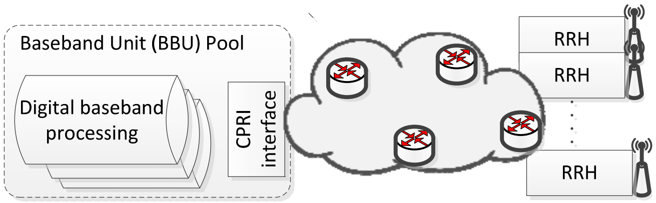


Figure 1. Cloud based Radio Access Networks (C-RAN) architecture

The organization of the paper is as follows: it starts by introducing the background on C-RAN. Afterwards we present related work with respect to time sensitive networks and packet preemption technology. Then, in Section IV, we describe the protocol based on the preemptive queuing system model. Section V includes numerical analysis followed by the presentation of the network simulation that validates the performance from the mathematical modeling. Section VI concludes the paper.

II. CLOUD RADIO ACCESS NETWORK (C-RAN)

The recent introduction of the C-RAN enables the geographical splitting between the Remote Radio Heads (RRHs) and the baseband processing units, which are originally integrated into one device. As shown in Figure 1, the Baseband Units (BBUs) from multiple base stations are pooled into a centralized and virtualized BBU pool. The front-haul network in C-RAN refers to the transport network between the RRHs and the BBU pool, where time-sensitive data and control messages are exchanged [2] [3].

In the C-RAN architecture, the main functions of a traditional base station can be divided into the radio functionalities and the baseband processing functionalities. The antenna module is responsible for power amplifier, frequency filtering and digital processing. The baseband module includes functions such as coding, modulation and Fast Fourier Transform (FFT), etc. Multiple BBUs are placed in a centralized location in order to enable a flexible utilization of BBUs resources and to reduce the operation and maintain cost. The common interface protocol between the RRHs and BBUs is the Common Public Radio Interface (CPRI), which carry transport and synchronization information from BBU to RRH.

The centralization and virtualization of BBUs resources provide benefits in terms of 1) flexible network utilization to cope with the irregular traffic distribution; 2) reduced deployment cost and power consumption gained from a central location; 3) enhanced cooperative decision making among multiple base station units and small cells.

All the advantages of C-RAN mentioned above cannot be achieved before a series of technical challenges can be addressed and solved [4].

- The expected bandwidth on the front-haul link is increased due to both the overhead generated from the RRH and BBU separation and the converged traffic to the centralized BBU pool.
- The expected latency and jitter requirements become stringent as the smallest as 100-250 μs depending on the function splits between the baseband processing and radio frequency functionalities.
- The inter cell interference among multiple small cells arises and should be minimized or used constructively.
- The current generation CPRI deployment is less than optimal solution due to its constant bit rate.
- The synchronization and timely delivery of traffic need to be ensured for mobile network operation.

The C-RAN front-haul network can be implemented based on either an optical transport solution or the traditional Ethernet network. Compared with the capacity-rich optical solution, the Ethernet-based front-haul network obtains popularity due to the widely spread Ethernet network anywhere. Reuse of existing network infrastructure brings benefits not only on saving deployment cost but also on keeping the consistence and continuity of the standards.

In the C-RAN front-haul network, the intermediate wireless signal needs to be transmitted between the BBU and RRH. The transmission has a strict delay constrain. The legacy Ethernet network technologies are not suitable for direct application in the front-haul network due to the lack of support for precise timing synchronization, low delay and latency and high throughput. Currently different active projects are formed under the umbrella of IEEE 802.1 TSN task group in order to tackle these difficulties for TSN applications. For example, IEEE 802.1as is available for the timing and synchronization. Based on the IEEE 802.1 Qbu standard, this paper presents the implementation of the packet preemption technology and evaluates the performances of the Ethernet based front-haul networks [5].

III. IEEE Time Sensitive Networking (TSN)

IEEE 802.1 TSN, formerly named the IEEE Audio Video Bridging (AVB), aims to define the specifications that allow time-synchronized low-latency streaming services [6]. Low delay and jitter requirements have been stringent phenomena for real-time applications. The standards target the requirements for the industrial applications, vehicle control services, control or streaming data in the local area networks, and so on.

The TSN traffic is classified into 4 classes, as listed below in Table I [7]. The class Control Data Traffic (CDT) has the highest priority and is intended to carry the control messages. The class A and class B are used to transport audio and video streams, respectively. Class BE handles the best effort traffic, such as the legacy Ethernet traffic, with no restriction on QoS.The traffic specifications consist of two main categories: the maximum frame size and the minimum frame interval. The maximum frame size indicates the packet size of source data. The minimum frame interval indicates the frequency of receiving data. Based on the application, each class is specified with delay and jitter constraints.

Regarding the fronthaul in a C-RAN network, strict requirements are defined on the links between the RRH and the BBU. These requirements such as clock synchronization and latency have to be satisfied. In this paper, simulations of TSN functions have been performed to combine TSN features in a Fronthaul network. The class CDT traffic is evaluated with reduced latency.

IV. Packet Preemption

In this section, we briefly describe the salient features of the packet preemption technology standardized in the IEEE 802.1 Qbu and IEEE 802.3br documents [8]. The technology uses the preemptive-resume queueing discipline. The aim is to ensure a deterministic behavior with low delay for time critical packet frames.

In the packet preemption standard, the types of traffic on an ingress port are classified into two groups: express traffic and preemptive traffic. The express traffic is used for the transmission of the class CDT data while the preemptive traffic is sent with other classes of traffic.

The Head-of-Line (HOL) problem is well known from the traditional FIFO queue discipline. This is solved by prioritizing packets in different queues. An express packet is transmitted before the queued preemptive packets. However, an express packet can still experience excessive delay, since the preemptive packet that started ahead can be a large size packet. The motivation of the packet preemption technology is to eliminate the waiting delay caused by ongoing transmission of a preemptive packet.

Figure 2 presents the different operations between the usual priority queuing and the preemptive queueing when a new packet arrives. The transmission of a preemptive packet can be suspended in order to allow one or multiple express packets to be transmitted. Afterwards, the remainder preemptive packet resumes transmission. It is notable that one preemptive packet can be preempted and resumed for several times. This provides the capabilities of a network switch to support a deterministic time control application.

Table I: TSN TRAFFIC TYPES AND REQUIREMENTS

Traffic types	Maximum Frame Size (bytes)	Minimum Frame Interval (us)
Class CDT	128	500
Class A	256	125
Class B	256	250
Class BE	256	None

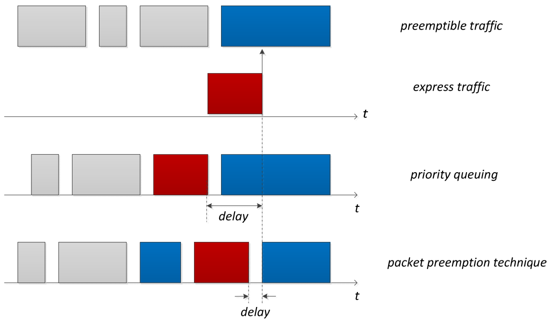


Figure 2. Priority queueing and the packet preemption queueing

To provide service differentiation, separated queues are applied to classify traffic into groups, and the packet preemption technique is used for priority scheduling at the switch node. In the traffic filtering and classifier module, the incoming data are classified into the express queue and the preemptive queue. In the transmission processing module, the system monitors the appearance of the express traffic. The preemption happens when the corresponding express traffic arrives. The preemption procedure occurs with some conditions stated in the standard. For example, the packet size is at least 64 bytes that remain to be transmitted. With this scheme, an express packet can take over the low priority traffic, even during the transmission. A format of *mPacket* is defined in the standard containing a complete packet (e.g., an express packet) or a continuation fragment of a preemptive packet. The detailed procedure of the preemption is shown in Figure 3.

To implement the 802.1Qbr, both transmitter and receiver switch should enable the packet preemption support as a TSN-enabled switch. In the transmitter side, the Ethernet frames are differentiated and classified into the express queue and the preemptive queue, respectively. When an express packet is received in the system while a preemptive packet is being processed, the express packet is processed immediately upon arrival assuming the packet preemption condition is fulfilled. The newly generated packet is formatted as *mPacket*, which carries the express packet, the complete or fragmented preemptive packet. A preemptive packet is interrupted and fragmented as a series of the continued fragments. An *mPacket* containing a continuation fragment of a preemptive packet has a fragment counter. The receiver identifies the packets and reassembles an incomplete preempted packet.

V. Statistical Model

Our goal is to analyze the performance of a front-haul network with TSN enabled switches, taking into consideration queuing and packet preemption in each node. In this section, we first introduce the statistical model to study the queuing delay based on a preemptive resume queuing model. In this part, we simplify the traffic model as Poisson arrivals and fixed size packets.

A single server system with limited queue size is considered, which has job classes of multiple priorities. The priority queue can have either non-preemptive or preemptive strategies. In a non-preemptive system, a job in service is not interrupted, even if a job of higher priority arrives and enters the queue. In a preemptive case, the service of an ongoing job will be interrupted by the new arrival of higher priority. The preemptive resume model means that the interrupted job from lower priority continues, when the higher priority job finishes.

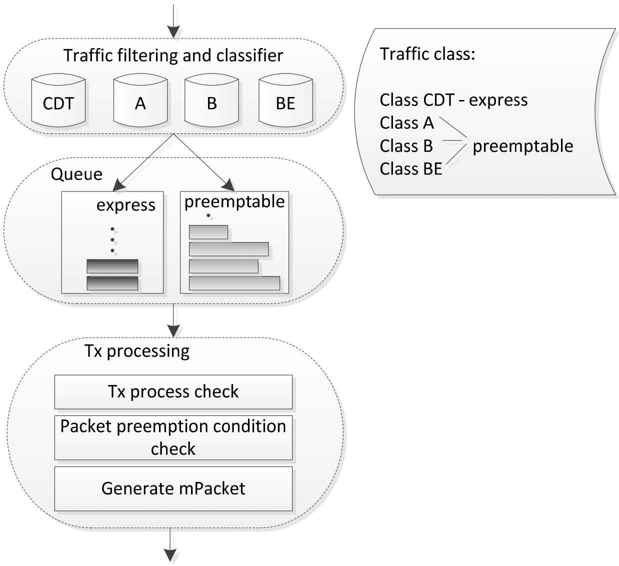


Figure 3. Diagram of packet preemption procedure

To simplify the analysis, the M/M/1/K model is considered, with one single server and a limited number of waiting positions. The data arrivals are Poisson distributed with rate λ. Two classes of traffic arrival to the system are: express traffic with arrival intensity of λ_e, and the pre-emptible traffic with arrival intensity of λ_p. The express traffic has a higher priority.

For each class, the service time is exponentially distributed with a mean value of S. We denote the offered traffic of each type as $A_e = \lambda_e * S_e$ and $A_p = \lambda_p * S_p$, where the jobs in the express and pre-emptible class are assumed with a mean service time S_e and S_p, respectively.

A. *Non-preemptive queuing model*

The numerical analysis for the queuing delay for each traffic class, express and preemptive traffic has been discussed in details in [9]. For the express traffic, the highest priority, waits until the service in progress is completed and waits for the existing jobs in the same queue. The mean waiting time W_e is calculated as:

$$W_e = V_{\mathrm{e}} + A_e \cdot W_e \qquad (1)$$

Where V_{e} is the mean residual service time of the current job under service, both express and preemptive traffic classes are considered.

For the low priority class (referring to the preemptive traffic), the mean waiting time, W_p, considers not only the remainder process and the already arrived jobs from the same and higher priority, but also the new arriving jobs with higher priority during the waiting time.

$$W_p = V_{e,p} + \cdot A_p \cdot W_p + A_e \cdot W_p \qquad (2)$$

B. Preemptive-resume queuing model

With preemptive resume property, a job with low priority is interrupted by the arrival of a higher priority job. The transmission will be continued from the point that it is interrupted later.

As the highest priority, the express traffic experiences only the expected remaining service time due to the existing jobs in the same queue, since with preemptive property the express traffic is not disturbed by lower priorities. Therefore, the mean waiting time W_e is same as (1). (but (1) includes low priority traffic under service- this traffic is preempted, in this case).

For the preemptive traffic, the mean waiting time considers the existing express traffic, which is already in the queueing system. Moreover, the extra waiting period caused by the interruption from the express traffic during the service time and the waiting time should be taken into account.

$$W_p = \frac{V_{e,p}}{1 - A_e} + \{W_p + s_p\} \cdot A_e \qquad (3)$$

C. Probability model

By analyzing the statistical queueing model, we can derive the mean waiting time for the express and the preemptive traffic. By using the state transition diagram of the Markov chain and presenting the state balance equations, we can derive the delay probability of the system. We model the number of queuing places used by each class in a switch as a continuous-time Markov chain.

The problem is illustrated with a simplified M/M/1/2 queue, where only one queueing place is allowed. The state (*i, j*) describes the number of express traffic, *i*, and the number of preemptive traffic, *j*, in the system. In the non-preemptive model, as shown Figure 4, the job with a low priority is under processing, while the high priority traffic is waiting, as shown in the (1, 1) state. 0 presents the state transition diagram for the preemptive priority queue model. Different to the non-preemptive case, when a new express traffic arrives, the service of the lower class traffic is stopped and the process of the express traffic starts, from state (0, 1) to (1, 1).

Recall the Markov property which states that the future process is only influenced by the current state of the process. Note that the probability of being in state (*i, j*) is *P(x,y)*. From the state transition diagram in a non-preemptive model in Figure 4, we obtain the following expression, Eq(4):

$$P(x,y) = \frac{\dfrac{A_e^x}{x!} \cdot \dfrac{A_p^y}{y!}}{\displaystyle\sum_{i=0}^{2} \sum_{j=0}^{2-i} \dfrac{A_e^i}{i!} \cdot \dfrac{A_p^j}{j!}} \qquad (4)$$

Where $A_{\mathrm{p}}^{\mathrm{x}} / x!$ and $A_{\mathrm{p}}^{y} / y!$ represent the state probability of one dimensional truncated Poisson distribution for the preemptive and express traffic, respectively.

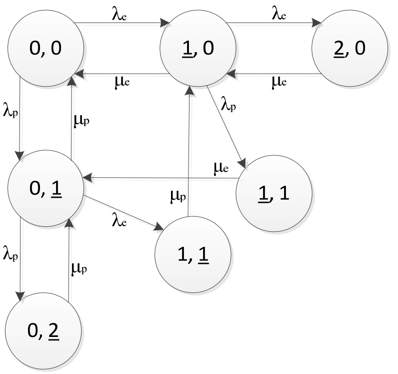


Figure 4. State transition diagram for M/M/1/2 non-preemptive priority queue

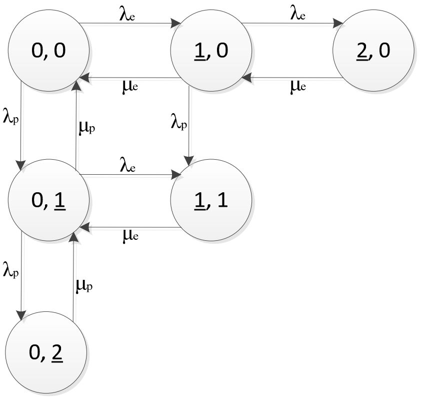


Figure 5. State transition diagram for M/M/1/2 preemptive priority queue

In non-preemptive model, the delay probability of express packet is supposed to contain *P(0,1)* where one preemptive packet is under service, thus the delay probability is estimated as:

$$\mathrm{D} = \sum_{i=0}^{1} \sum_{j=0}^{1-i} \mathrm{P}(i,j) \qquad (5)$$

By considering the Markov chain in 0, the processing of the preemptive traffic is interrupted when the express traffic arrives in the state (1, 1). From (4), we can estimate the threshold of queuing delay. The preemption will be performed and the delay for each class is increased by controlling the arrival rate. The result can be extended to a switch with a large queue size.

VI. PERFORMANCE SIMULATION STUDY AND RESULTS

To evaluate the packet preemption technology and its behaviors, a TSN enabled Ethernet switch is examined by simulations. The simulation scenarios are setup in Riverbed Modeler[10]. Independent simulation was performed with various random seed numbers. Both the traditional priority queuing model and the preemptive resume queuing model are implemented. Consequently, the simulation will provide information about the limiting factors and the performance metrics of delay, packet loss and throughput.

We consider two types of traffic in the communication system, express traffic and preemptive traffic. We measure the performance based on various traffic intensities, packet sizes, and ratio between different types of traffic.

A. *Queuing delay vs. traffic intensity*

The relative load between the express traffic and the preemptive traffic is varied from 0.1 to 1. Figure 6 shows the delay of two classes under different ratios. The increasing percentage of the express traffic introduces delay increment on the preemptive traffic. The delay for the express traffic keeps a low value in both the packet preemption and non-packet preemption cases. It is observed that the packet preemption reduces the queueing delay for the express traffic.

B. *Queuing delay vs. packet length*

We evaluate the influence of the packet size to the queueing delay. In this scenario, the input express traffic takes up 50 percent of the preemptive traffic. The packet size of the express traffic is fixed as a uniform distribution of 128 bytes. The packet length of the preemptive traffic is varied from 128 bytes to 1024 bytes. With the packet preemption technique, the queueing delay of the express traffic is increased, when the preemptive packets are mostly in small size. This is due to the rule of packet preemption, which examines the remaining size to be at least 64 bytes. When the preemptive packets are small, the chance of packet preemption is reduced. The express traffic has to wait for the ongoing transmission as in the non-packet preemption case. As shown in Figure 7, the smaller the packet size of the preemptible traffic, the fewer chances to segment the preemptible packets. Therefore, the express traffic has to be queued and waited for the preemptible traffic.

VII. CONCLUSION

In this paper, we analyzed the packet preemption scheme for reducing the waiting time in the queue and supporting service differentiation in the cloud radio access networks CRAN. The packet preemption technique favors the time-sensitive data by interrupting the interfered traffic and reducing the waiting time. The numerical analysis (not really shown) and the simulation results showed that the delay for the time-sensitive data is reduced dramatically. The influence of the traffic volume and packet length regarding the delay was analyzed. Packet preemption thus proved as an effective method to support time sensitive traffic over C-RAN fronthaul networks.

ACKNOWLEDGMENT

This work receives support from the “intelligent 5G mobile Ethernet Radio Access Network (ERAN)” project founded by Innovation Fond Denmark.

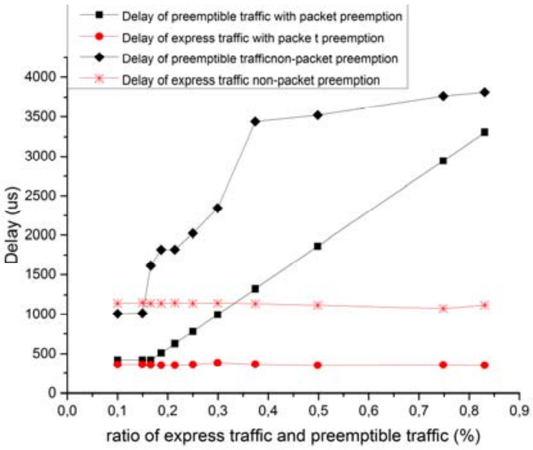


Figure 6. Queueing delay under different traffic loads

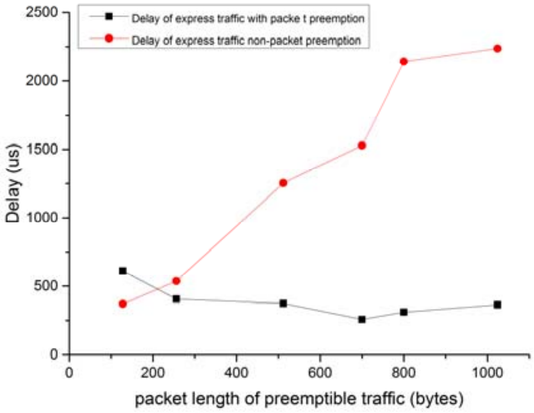


Figure 7. Queueing delay under different packet length

REFERENCES

[1] Time Sensitive Networking Task Group, http://www.ieee802.org/1/pages/tsn.html [retrieved: August 2017].

[2] C-RAN The Road Towards Green RAN. Tech. rep. China Mobile Research Institute, October 2011.

[3] Y. Lin, L. Shao, Z. Zhu, Q. Wang and R. K. Sabhikhi, "Wireless network cloud: Architecture and system requirements," in IBM Journal of Research and Development, vol. 54, no. 1, pp. 4:1-4:12, January-February 2010. doi: 10.1147/JRD.2009.2037680.

[4] A. Checko et al., "Cloud RAN for Mobile Networks—A Technology Overview," in IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 405-426, Firstquarter 2015. doi: 10.1109/COMST.2014.2355255.

[5] IEEE P802.1CM Draft standard for local and metropolitan area networks – Time sensitive networks for Fronthaul, October 2016.

[6] G. A. Ditzel and P. Didier, “Time Sensitive Network (TSN) Protocols and use in Ethernet/IP systems”, ODVA Industry conference & 17th Annual meeting, October 2015.

[7] S. Thangamuthu, N. Concer, P. J. L. Cuijpers and J. J. Lukkien, "Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications," 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, 2015, pp. 55-60. doi: 10.7873/DATE.2015.0045.

[8] IEEE P802.3br Draft Standard for Ethernet Amendment: Specification and Management Parameters for Interspersing Express Traffic, January 2016.

[9] V. B. Iversen, “Teletraffic engineering and network planning”, Publisher: DTU Fotonik, 2015.

[10] OPNET Technologies – Network Simulator, Riverbed. www.riverbed.com.