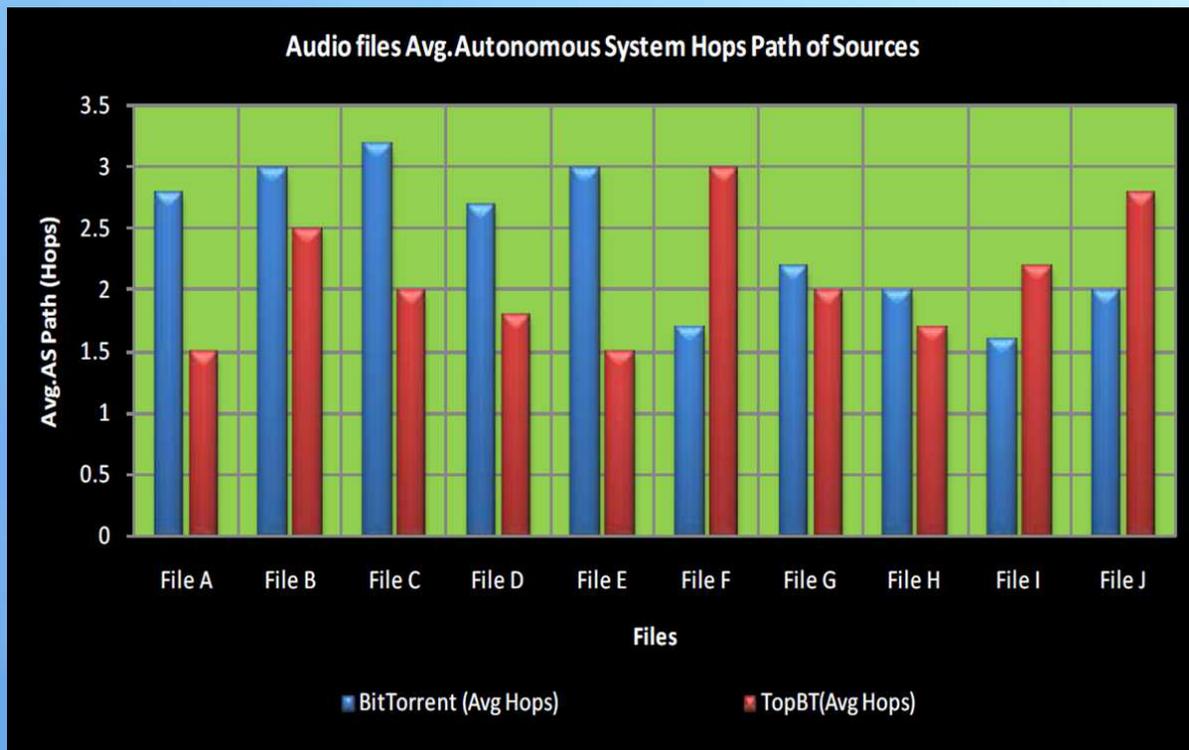


Communications and Network



ISSN: 1949-2421



Journal Editorial Board

ISSN Print: 1949-2421 ISSN Online: 1947-3826

<http://www.scirp.org/journal/cn>

Editor-in-Chief

Prof. Bharat Bhargava Purdue University, USA

Editorial Board

Prof. Ai Chen Chinese Academy of Sciences, China
Dr. Xiaochun Cheng Middlesex University, UK
Prof. Nagy Ibrahim Elakalashy Minoufiya University, Egypt
Prof. Abdullah Eroglu Indiana University-Purdue University Fort Wayne, USA
Dr. Jordán Pascual Espada University of Oviedo, Spain
Prof. Xiaopeng Fan Harbin Institute of Technology (HIT), China
Prof. Cunqing Hua Shanghai Jiao Tong University, China
Dr. Winai Jaikla King Mongkut's Institute of Technology Ladkrabang, Thailand
Prof. Shanshan Jiang Broadcom Corporation, USA
Prof. Xiaohong Jiang Tohoku University, Japan
Prof. Rami Langar University of Pierre and Marie Curie, France
Prof. Yu Liu Qualcomm Incorporated, USA
Prof. Hussein Mouftah University of Ottawa, Canada
Prof. Ruy de Oliveira Institute of Technology of MT, Brazil
Prof. Lu Peng Louisiana State University, USA
Prof. Eng. Sattar B. Sadkhan Babylon University, Iraq
Dr. Karan Singh Gautam Buddha University, India
Prof. Xiaojun Wang Dublin City University, Ireland
Dr. Jonathan Wellons Vanderbilt University, USA
Dr. Gang Yao University of Illinois at Urbana Champaign, USA
Dr. Hsiang-Fu Yu National Taipei University of Education, Chinese Taipei
Prof. Francesco Zirilli Sapienza Università di Roma, Italy

Table of Contents

Volume 7 Number 1

February 2015

| | |
|--|----|
| Control Access Point of Devices for Delay Reduction in WBAN Systems with CSMA/CA | |
| A. Nemoto, P. T. Hiep, R. Kohno..... | 1 |
| Performance Study of Locality and Its Impact on Peer-to-Peer Systems | |
| M. Y. Talha, R. W. Aldhaferi, M. H. Awedh..... | 12 |
| Improving Queuing System Throughput Using Distributed Mean Value Analysis to Control Network Congestion | |
| F. Shahzad, M. F. Mushtaq, S. Ullah, M. A. Siddique, S. Khurram, N. Saher..... | 21 |
| Hard Decision-Based PWM for MIMO-OFDM Radar | |
| O. Daoud..... | 30 |
| Dynamic Capacity Allocation in OTN Networks | |
| M. C. Taful, J. Pires..... | 43 |
| High-Level Portable Programming Language for Optimized Memory Use of Network Processors | |
| Y. Kanada..... | 55 |

Communications and Network (CN)

Journal Information

SUBSCRIPTIONS

The *Communications and Network* (Online at Scientific Research Publishing, www.SciRP.org) is published quarterly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: \$89 per issue.

To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements

Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.

E-mail: sub@scirp.org

COPYRIGHT

COPYRIGHT AND REUSE RIGHTS FOR THE FRONT MATTER OF THE JOURNAL:

Copyright © 2015 by Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>

COPYRIGHT FOR INDIVIDUAL PAPERS OF THE JOURNAL:

Copyright © 2015 by author(s) and Scientific Research Publishing Inc.

REUSE RIGHTS FOR INDIVIDUAL PAPERS:

Note: At SCIRP authors can choose between CC BY and CC BY-NC. Please consult each paper for its reuse rights.

DISCLAIMER OF LIABILITY

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assume no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: cn@scirp.org

Control Access Point of Devices for Delay Reduction in WBAN Systems with CSMA/CA

Akinobu Nemoto¹, Pham Thanh Hiep², Ryuji Kohno²

¹Department of Medical Informatics, Yokohama City University, Yokohama, Japan

²Division of Physics, Electrical and Computer Engineering, Yokohama National University, Yokohama, Japan

Email: anemoto@med.yokohama-cu.ac.jp, hiep@ynu.ac.jp, kohno@ynu.ac.jp

Received 10 January 2015; accepted 26 January 2015; published 29 January 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Due to the gathering of sickrooms and consultation rooms in almost all hospitals, the performance of wireless devices system is deteriorated by the increase of collision probability and waiting time. In order to improve the performance of wireless devices system, relay is added to control the access point and then the access of devices is distributed. The concentration of access point is avoided and then the performance of system is expected to be improved. The discrete time Markov chain (DTMC) is proposed to calculate the access probability of devices in a duration time slot. The collision probability, throughput, delay, bandwidth and so on are theoretically calculated based on the standard IEEE802.15.6 and the performance of the system with and without relay is compared. The numerical result indicates that the performance of the system with control access point is higher than that of the system without control access point when the number of devices and/or packet arrive rate are high. However, the system with control access point is more complicated. It is the trade-off between the performance and the complication.

Keywords

Standard IEEE802.15.6, Discrete Time Markov Chain Method, Control Access Point, Bandwidth Efficiency, Delay

1. Introduction

1.1. The Problem of WLAN in Hospitals

In almost hospitals, sickrooms and consultation rooms are respectively gathered at one place for convenience of patients. It may be good for patients and hospital sites, however, on the view point of wireless system, there is a problem. Since medical devices access the wireless local area network (WLAN) base station via wireless chan-

nel, the collision when more than one devices access the channel in the same time, will occurs depending on the number of devices and the number of data packets that be generated by every device in one second. Moreover, a lot of devices access the WLAN base station that is close to the consultation rooms (Wireless LAN 2 in **Figure 1**), whereas a few devices access the WLAN base station that is far from consultation rooms (Wireless LAN 1). The access of devices concentrates at Wireless LAN 2, consequently, the probability of collision increases, and then the throughput decreases, the delay increase. As a result the bandwidth efficiency decreases.

1.2. Aims and Motivations

Since many body functions are traditionally monitored and separated by a considerable period of time, it is hard for doctors to know what is really happening. This is the reason why the monitoring of movement and all body functions in daily life are essential. The delay of patients' data as well as the collision of data packets may let doctors misunderstand and information data be lost by timeout. In order to decrease the delay and increase the throughput, the relay can be set to avoid the concentration of WLAN base station. As shown in **Figure 1**), some devices assess the wireless LAN 1 via the relay, therefore, the number of devices that access the wireless LAN 2 is reduced, and then the bandwidth efficiency is expected to be higher. However, the delay due to signal processing at relay should be considered. At scheme 1, all devices access the wireless LAN 2, whereas at scheme 2, the relay is set and devices access the channel via either wireless LAN 1 or 2. The performance of both schemes 1 and 2 is mathematically analyzed base on standard IEEE802.15.6. The throughput, delay and bandwidth efficiency of both schemes are numerically compared.

1.3. Related Works

According to an emergency of wireless body area network (WBAN), the standard IEEE802.15.6 was established in Feb. 2012 [1]. An overview of the standard and performance analyses of WBAN based on bandwidth efficiency and delay were represented in [2]-[4]. In these papers, however, the WBAN is assumed to consists of only one device that keeps transmitting a data packet. Packet arrival rates and collisions due to transmission of multiple devices in the same time weren't considered. On the other hand, a Physical layer (PHY), Media Access Control (MAC) layer and network layer of WBAN were researched in [5] [6]. Furthermore, the control on MAC layer was analyzed to improve the performance of WBANs [7] [8]. The transmission of implanted devices was considered under conditions of low transmit power and low harmful influence on a human body [9] [10]. The performance of WBANs that has multiple devices and multiple user priorities were analyzed in both saturation [11] [13] [14] and non-saturation [12]. Additionally, WBANs were analyzed in further detail when a superframe with beacon mode and an access phases length were taken into consideration in [13] [14], respectively. However, efficiencies of number of devices, packet arrival rates, packet sizes, etc. on the throughput of each device and the total throughput, the delay and the bandwidth efficiency of system hasn't been discussed.

1.4. Organization of the Paper

The rest of paper is organized as follows. We introduce a brief of PHY and MAC layers of standard IEEE802.15.6

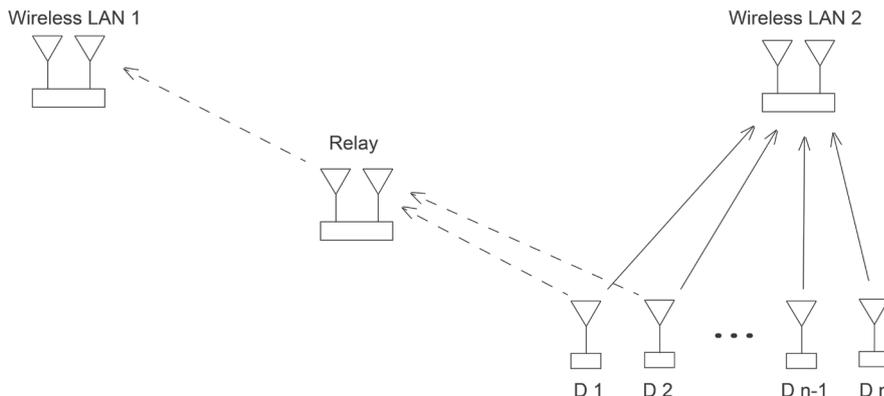


Figure 1. The wireless LAN system in a hospital.

in Section 2. The discrete time Markov chain is proposed and then the performance of both schemes 1 and 2 with CSMA/CA is analyzed in Section 3. The numerical evaluation of both schemes is described and compared in Section 4. Finally, Section 5 concludes the paper.

2. Brief of Standard IEEE802.15.6

A brief of the standard that related to our research is described in this section. The further detail of standard can be found in [1] [2].

2.1. PHY Layer

The IEEE802.15.6 defines three different PHYs, *i.e.*, human body communication (HBC), narrowband (NB) and ultra wideband (UWB). Furthermore, the NB is divided in several frequency bands and a data rate, symbol rate, etc. of every frequency band are different. We analyze the system in 2400 MHz - 2483.5 MHz band as an example, the analysis in different frequency band is similar. The physical protocol data unit (PPDU) of NB PHY is described in **Figure 2**. Components of PPDU are fixed, excepted the payload. Parameters of PHY layer are summarized in **Table 1**.

2.2. MAC Layer

The algorithm of CSMA/CA based on IEEE802.15.6 is described as follows. All devices set their backoff counter

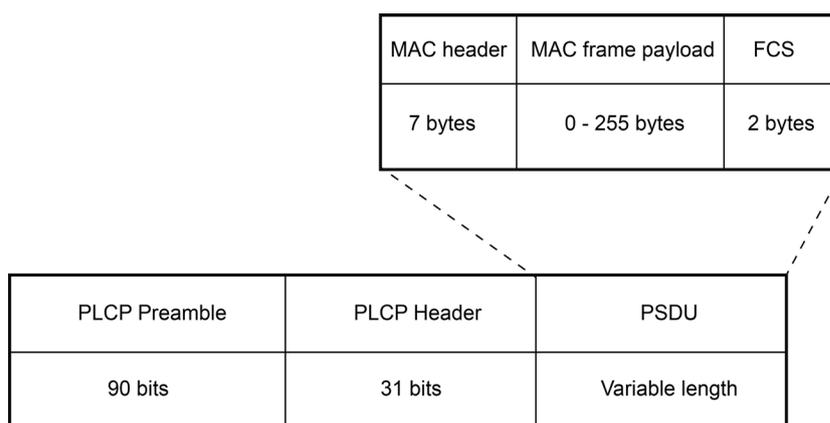


Figure 2. PPDU of NB PHY.

Table 1. Parameters of PHY layer.

| Frequency band [MHz] | 2400 - 2483.5 |
|--|---------------|
| Packet component | PSDU |
| Modulation DBPSK | DBPSK |
| Symbol rate R_s [ksps] | 600 |
| Data rate R_{hdr} [kbps] | 242.9 |
| Clear channel assessment [bits] | 63 |
| MAC header [bits] | 56 |
| MAC footer [bits] | 16 |
| CSMA slot time T_s [μ s] | 125 |
| Short interframe spacing time T_{pSTFS} [μ s] | 75 |
| Preamble [bits] | 88 |
| Propagation delay α [μ s] | 1 |

to a random integer number uniformly distributed over the interval $[1, W]$, where W is a contention window within $(W_{\min}; W_{\max})$. The value of W_{\min} and W_{\max} varies depending on the user priorities (UPs). However, in this paper, the UP of all devices is assumed to be the same as zero-th UP. The extension for multiple UPs is straightforward.

As shown in **Figure 3**, a device starts decrementing its back off counter by one for each idle CSMA slot. When the back off counter reaches zero, the device transmits its packet. Once the channel is busy because of transmission of another device, the device locks the back off counter until the channel is idle. The transmission is failed if the device fails to receive an acknowledgement (ACK) due to a collision or being unable to decode. The W is doubled for even number of failures until it reaches W_{\max} . The maximum number of back off stages is bound by a retry limit m . Once the number of retries exceeds the predefined retry limit m , the packet is discarded. When the transmission is successful, the W is set to W_{\max} . The W of zero-th UP is represented in **Table 2**.

3. Performance Analysis of WBANs

3.1. Discrete Time Markov Chain

At first, the performance of scheme 1 is analyzed. The scheme 1 consists of a single base station, the wireless LAN 2, and n devices in a star topology, $D1, D2, \dots, Dn$ (**Figure 1**). All devices can access the wireless LAN 2 directly, however, the wireless LAN 1 is out of them range. The discrete time Markov chain (DTMC) is proposed to calculate the access probability of each device in every time slot. The proposal DTMC of device i with empty state is described in **Figure 4** and notations used in this section are listed in **Table 3**. A packet arrival rate of all devices is assumed the same and denoted by λ . Hence, $\rho = 1 - e^{-\lambda Ts}$, where e denotes the Napier's constant, denotes the probability that the device has a packet to transmit in duration time of Ts . The transmission failed probability and the idle probability of device i are respectively expressed as

$$P_{i, fail} = P_{i, col} + PER_i,$$

$$P_{i, idle} = \frac{\prod_{k=1}^n (1 - \tau_k)}{1 - \tau_i}, \tag{1}$$

here $P_{i, col} = 1 - \prod_{k \neq i}^n (1 - \tau_k)$. The state transmission probabilities of DTMC method are represented as follows.

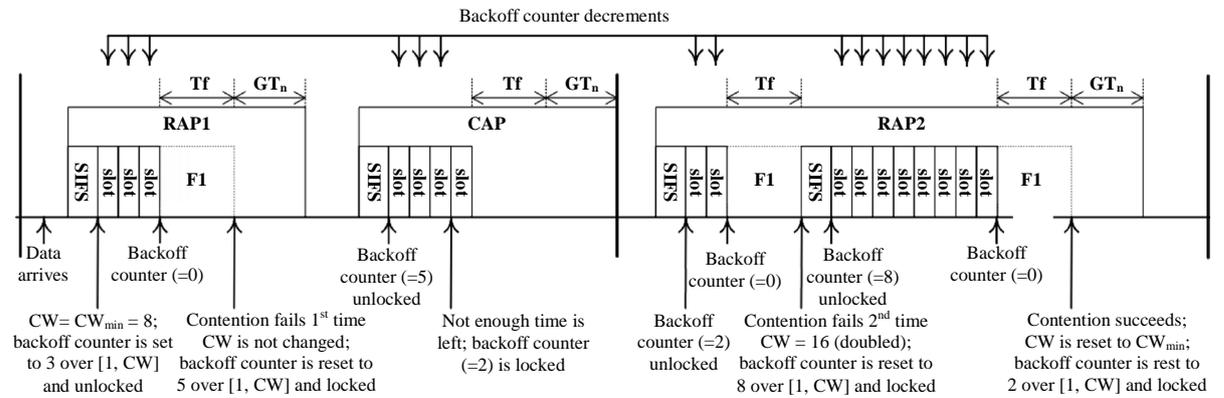


Figure 3. An example of operation of CSMA/CA and relationships of time durations.

Table 2. Contention window for every UP.

| Number of retransmissions | 0 | 1 | 2 | 3 | 4 and over |
|---------------------------|----|----|----|----|------------|
| W | 16 | 16 | 32 | 32 | 64 |

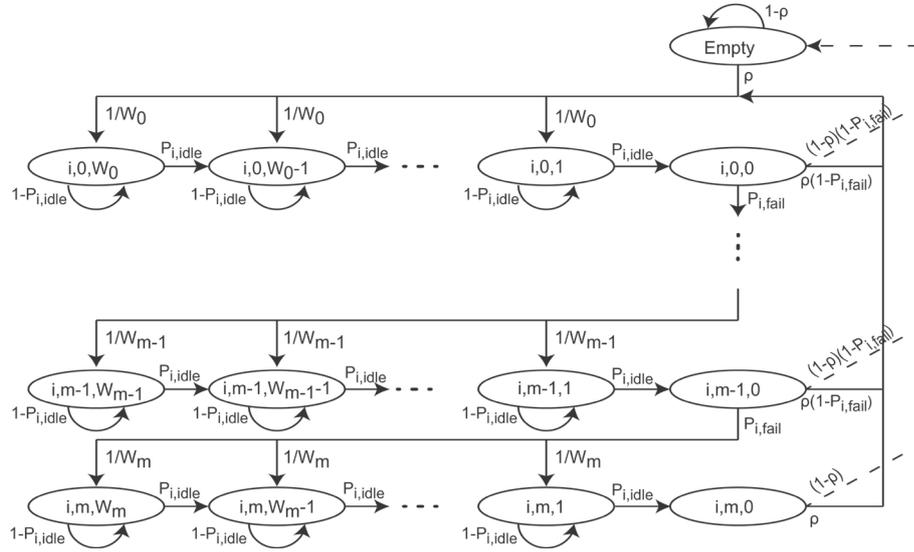

Figure 4. Algorithm of DTMC method.

Table 3. Explanation of notations.

| Notation | Explanation |
|--------------|--|
| λ | Packet arrival rate during a unit time |
| ρ | Packet arrival rate during a slot time |
| m | Packet retry limit |
| n | Total number of devices |
| x | Payload size |
| Total x | Total data |
| T_i | Access probability during a slot time |
| $b_{i,k,j}$ | Stationary distribution with backoff stage k , backoff counter j |
| $P_{i,idle}$ | Channel idle probability |
| $P_{i,fail}$ | Transmission failed probability |
| P_{icol} | Collision probability |
| PER_i | Packet error rate |
| W_k | Contention window of k backoff stage |

$$\Pr\{i, k, j | i, k, j+1\} = P_{i,idle}, \quad \text{for } k \in [0, m], j \in [0, W_k],$$

$$\Pr\{i, k, j | i, k-1, 0\} = \frac{P_{i,fail}}{W_k}, \quad \text{for } k \in [1, m], j \in [1, W_k],$$

$$\Pr\{i, k, j | i, k, j\} = 1 - P_{i,idle}, \quad \text{for } k \in [0, m], j \in [1, W_k],$$

$$\Pr\{i, 0, j | i, m, 0\} = \frac{\rho}{W_0}, \quad \text{for } j \in [1, W_0],$$

$$\Pr\{i, 0, j | \text{empty}\} = \frac{\rho}{W_0}, \quad \text{for } j \in [1, W_0],$$

$$\Pr\{\text{empty} | i, k, 0\} = (1-\rho)(1-P_{i,fail}), \quad \text{for } k \in [0, m-1],$$

$$\Pr\{\text{empty} | i, m, 0\} = 1-\rho,$$

$$\Pr\{\text{empty} | \text{empty}\} = 1-\rho.$$

(2)

As shown in **Figure 4**, we have

$$\sum_{k=0}^m \sum_{j=0}^{W_k} b_{i,k,j} + b_{\text{empty}} = 1 \quad (3)$$

Moreover, the stationary distribution can be calculated by using the state transition probability.

$$\begin{aligned} \sum_{k=1}^m b_{i,k,0} &= \frac{P_{i,\text{fail}} (1 - P_{i,\text{fail}}^m)}{1 - P_{i,\text{fail}}} b_{i,0,0}, \\ \sum_{j=0}^{W_0} b_{i,0,j} &= \frac{W_0 + 1}{2P_{i,\text{idle}}} b_{i,0,0}, \\ b_{\text{empty}} &= \frac{1 - \rho}{\rho} b_{i,0,0}. \end{aligned} \quad (4)$$

From above equations, the $b_{i,0,0}$ can be described as a function of $P_{i,\text{idle}}$, $P_{i,\text{fail}}$, ρ , W_k and T_i . Furthermore, the access probability of every device can be calculated by solving n equations.

$$\tau_i = \sum_{k=0}^m b_{i,k,0} = \frac{1 - P_{i,\text{fail}}^{m+1}}{1 - P_{i,\text{fail}}} b_{i,0,0} \quad (5)$$

3.2. System Throughput

The probability in which at least one device is sending a packet is called as transmission probability, P_{tran} .

$$P_{\text{tran}} = 1 - \prod_{j=1}^n (1 - \tau_j) \quad (6)$$

The successful probability of device i means that only device i is transmitting on the medium under condition on the fact that at least one device is transmitting and is represented by $P_{i,\text{suc}}$. In addition, the coordinator can decode the packet correctly.

$$P_{i,\text{suc}} = \frac{\tau_i \prod_{j=1}^n (1 - \tau_j)}{(1 - \tau_i) P_{\text{tran}}} (1 - \text{PER}_i) \quad (7)$$

Let $P_{\text{suc}} = \sum_{i=1}^n P_{i,\text{suc}}$ denote the total successful probability of all devices. Once the transmission is successful, the device receives a ACK packet with no payload from the coordinator, whereas the device receives NACK packet or nothing after the timing to receive the ACK packet if the transmitted packet is collided or unable to decode. Consequently, the duration time to transmit a packet successfully, T , is assumed to equal to the duration time of failed transmission, hereafter T is called as the successful transmission time. The successful transmission time is the total duration time to transmit a packet, includes the duration time to transmit a data packet (T_{data}), interframe spacing (T_{pSIFS}), ACK packet (T_{ACK}) and delay time (α).

$$T = T_{\text{DATA}} + T_{\text{ACK}} + 2T_{\text{pSIFS}} + 2\alpha \quad (8)$$

Let T_P , T_{PHY} , T_{MAC} , T_{BODY} and T_{FCS} denote the duration time to transmit a preamble, PHY header, MAC header, MAC body and FCS, respectively. Therefore, the duration time to transmit a data packet is given by

$$\begin{aligned} T_{\text{DATA}} &= T_P + T_{\text{PHY}} + T_{\text{MAC}} + T_{\text{BODY}} + T_{\text{FCS}}, \\ &= \frac{\text{Preamble} + \text{PHY header}}{R_s} + \frac{8(\text{MAC header} + x + \text{MAC footer})}{R_{\text{hdr}}}. \end{aligned} \quad (9)$$

Since an immediate ACK/NACK carries no payload, its transmission time is represented as follows.

$$\begin{aligned}
T_{\text{ACK}} &= T_p + T_{\text{PHY}} + T_{\text{MAC}} + T_{\text{FCS}}, \\
&= \frac{\text{Preamble} + \text{PHY header}}{R_s} + \frac{8(\text{MAC header} + \text{MAC footer})}{R_{\text{hdr}}}.
\end{aligned} \tag{10}$$

Finally, the throughput of device i is described as

$$\begin{aligned}
\text{Thro}_i &= \frac{P_{\text{tran}} P_{i,\text{suc}} x r}{(1 - P_{\text{tran}}) T_s + P_{\text{tran}} P_{\text{suc}} T + P_{\text{tran}} (1 - P_{\text{suc}}) T} \\
&= \frac{P_{\text{tran}} P_{i,\text{suc}} x r}{(1 - P_{\text{tran}}) T_s + P_{\text{tran}} T},
\end{aligned} \tag{11}$$

and the system throughput becomes

$$\text{Thro} = \sum_{i=1}^n \text{Thro}_i \tag{12}$$

The throughput of scheme 2 is also represented by (12). However, several devices access the channel via the relay and the wireless LAN 1, therefore, the concentration at the wireless LAN 2 is avoided and the successful probability of all devices increases. As a result, the throughput of system is expected to increase.

3.3. Delay

The average access delay D , defined as the time elapsed between the time instant when the frame is put into service and the instant of time the frame terminates a successful delivery. Under the assumption of no retry limits, this computation is straightforward. In fact, we may rely on the well known Little's Result, which states that, for any queueing system, the average number of customers in the system is equal to the average experienced delay multiplied by the average customer departure rate. The application of Little's result to our case yields:

$$D = \frac{x\lambda}{\frac{\text{Thro}}{n}} = \frac{x\lambda}{\text{Thro}_i} \tag{13}$$

The delay computation is more elaborate when a frame is discarded after reaching a predetermined maximum number of retries m . In fact, in such a case, a correct delay computation should take into account only the frames successfully delivered at the destination, while should exclude the contribution of frames dropped because of frame retry limit (indeed, the delay experienced by dropped frames would have no practical significance).

To determine the average delay in the finite retry case, we can still start from Little's Result, but we need to replace λ in (13) with the average number of frames that will be successfully delivered. Thus, (13) can be rewritten by

$$D = \frac{x\lambda\beta_i}{\text{Thro}_i} \tag{14}$$

here, β_i denotes the probability that a randomly chosen frame will be successfully transmitted before the retransmission reaches the retry limit. Therefore, the β_i is represented as follows.

$$\beta_i = P_{i,\text{suc}} \sum_{j=0}^{m-1} (1 - P_{i,\text{suc}})^j \tag{15}$$

For Scheme 2, the delay due to the multiple access at the wireless LAN 1 and 2 is similar to (14). However, the delay due to the capability of relay also should be considered. The delay due to the relay is calculated by $\frac{\sum_{j \in Q} \text{Thro}_j}{C}$, here Q denote the set of devices that access the relay and the C is the capability of relay. Therefore, the average delay of information data that is transmitted via relay is represented as follows.

$$D = \frac{x\lambda\beta_j}{\text{Thro}_j} + \frac{\sum_{j \in Q} \text{Thro}_i}{C} \quad (16)$$

The delay of scheme 2 is the maximal delay of information data that is transmitted to wireless LAN 1 and 2.

3.4. Bandwidth Efficiency

In order to compare the system with and without relay, the bandwidth efficiency is adopted. The bandwidth efficiency of both schemes 1 and 2 is calculated as the ratio of total throughput of system and the total generated data. Notice that the total throughput of scheme 1 and 2 is different.

$$\delta = \frac{\text{Thro}}{nx\lambda}. \quad (17)$$

4. Numerical Evaluation

The system model is the same as mentioned above and the parameters in **Table 1** are used. The average distance between all devices and the wireless LAN 1 and 2 is respectively 500 m and 250 m. The relay is set at halfway between the devices and the wireless LAN 1. The delay of propagation is taken into account. The capability of relay is assumed to be 300 Mbps. The noise-free is also assumed. At first, the performance of scheme 1 is illustrated.

The throughput of scheme 1 based on lambda and the number of devices is described in **Figure 5** and **Figure 6**, respectively. The generated data is the total data that is generated at all devices, however the generated data isn't always successfully transmitted due to the collision and the time out. Therefore, the throughput of system is considerably smaller than the generated data, especially when the number of devices and/or the lambda are high. Moreover, the delay of scheme 1 also increase when the number of devices and/or the lambda increase (**Figure 7**). These are the reason the scheme 2 is taken into consideration as description in Section 1.2.

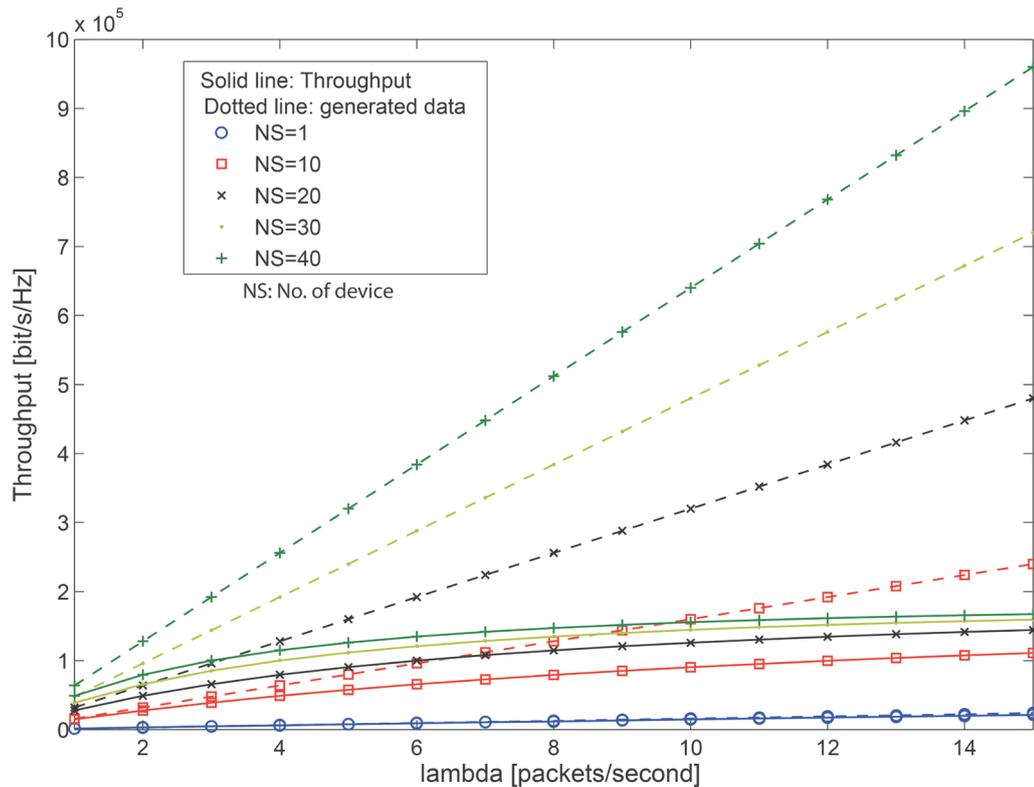


Figure 5. Throughput of scheme 1 based on lambda.

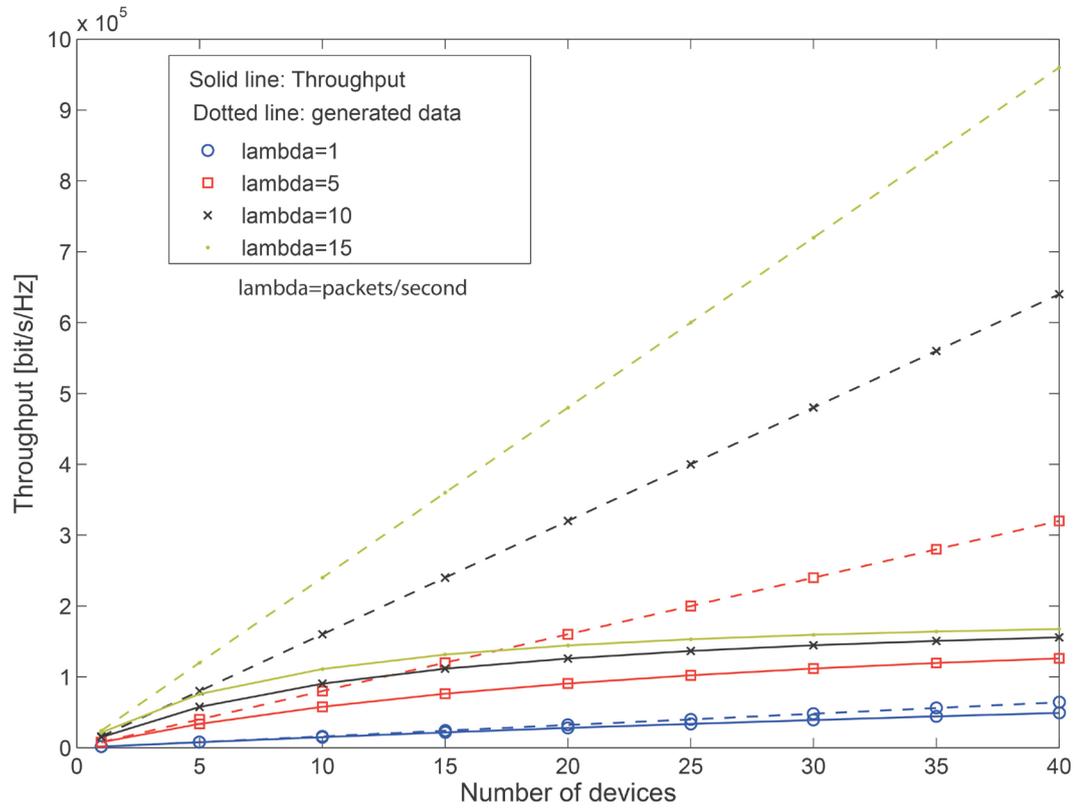


Figure 6. Throughput of scheme 1 based on the number of devices.

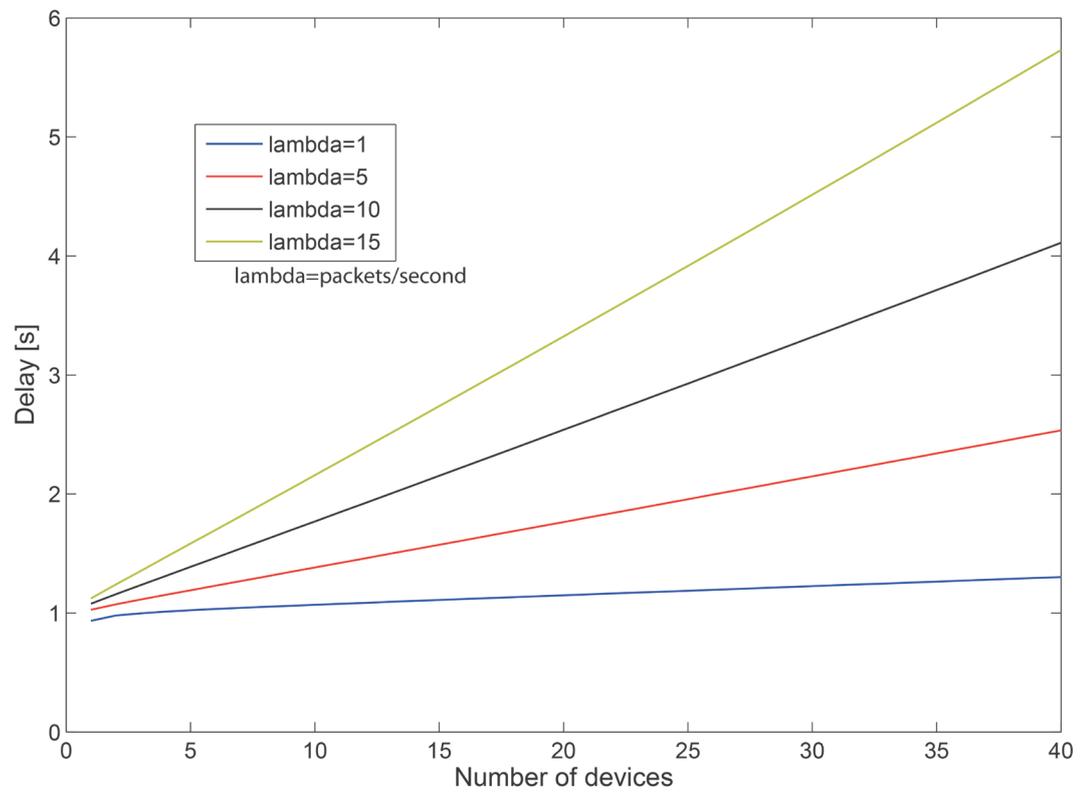


Figure 7. Delay of scheme 1.

The comparison on the delay and the bandwidth efficiency of both schemes 1 and 2 is respectively described in **Figure 8** and **Figure 9**, where the number of devices is fixed to be 10 and 40. For scheme 2, since the concentration at the wireless LAN 2 is avoided, the collision probability decreases. Therefore, the throughput of system increase and then the delay as well as the bandwidth efficiency increase and be higher than that of scheme 1, especially when the number of devices and/or the lambda are large. When the number of devices and the lambda are low, the difference of schemes 1 and 2 is small. Notice that the scheme 2 is more complicated due to the adding of relay and controlling the transmission of devices. It means that there are the trade off between the performance and the complication of scheme 2.

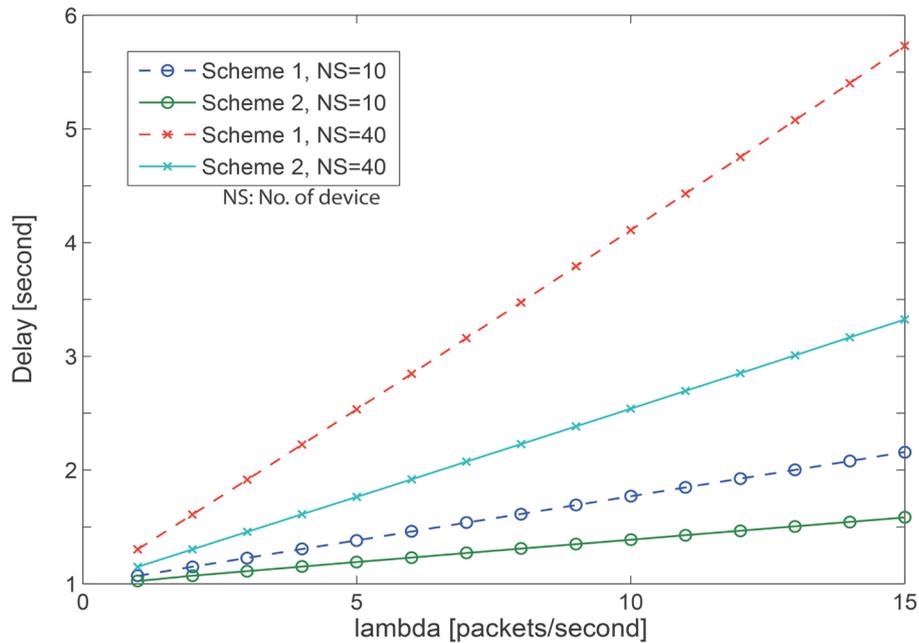


Figure 8. Delay base on lambda.

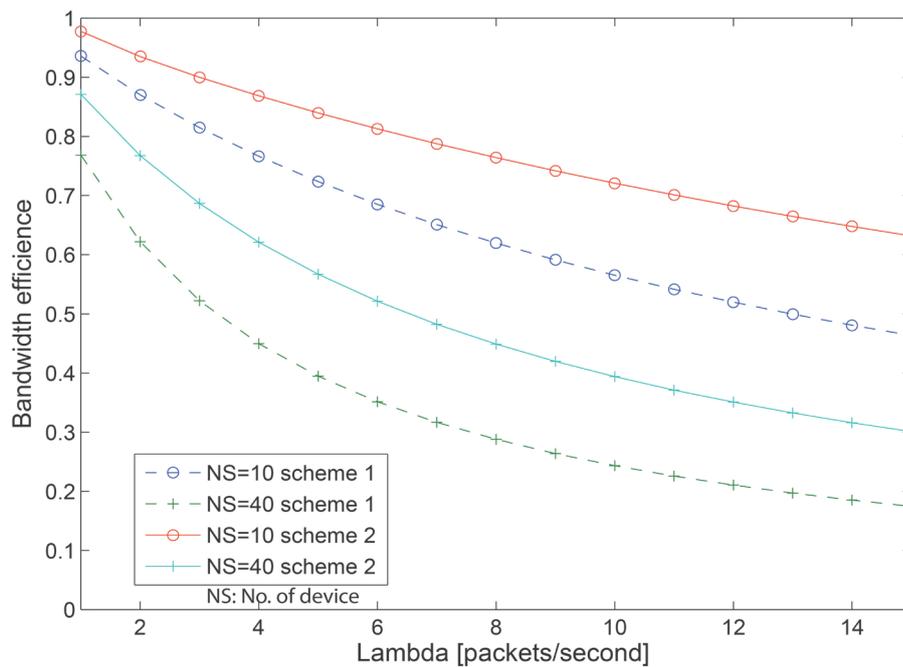


Figure 9. Bandwidth efficiency.

5. Conclusions

The wireless system in a hospital has been taken into consideration and the performance is analyzed based on the standard IEEE802.15.6. The DTMC method was proposed to calculate the access probability and then the collision probability, the successful probability, the throughput, the delay, the bandwidth efficiency of system have been theoretically calculated. The performance of system with and without relay is also numerically compared, the bandwidth efficiency of scheme 2 is higher while the delay is smaller than that of scheme 1 when the number of devices and/or the packet rate are large. However, the scheme 2 is more complicated due to the controlling transmission of devices and the adding of relay.

The devices were assumed to transmit the information data to either the wireless LAN 1 or the wireless LAN 2. However, the control method hasn't been explained clearly. Moreover, the CSMA/CA was adopted, another access protocol [15] wasn't taken into account leave them to our future works.

References

- [1] Wireless Personal Area Network Working Group (2012) IEEE Standard 802.15.6, Wireless Body Area Networks. *IEEE Standards*, 1-271.
- [2] Kwak, K.S., Ullah, S. and Ullah, N. (2010) An Overview of IEEE 802.15.6 Standard. *Proceedings of 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Rome.
- [3] Martelli, F., Buratti, C. and Verdone, R. (2011) On the Performance of an IEEE 802.15.6 Wireless Body Area Network. *Proceedings of European Wireless 2011*, Vienna.
- [4] Ullah, S., Chen, M. and Kwak, K.S. (2012) Throughput and Delay Analysis of IEEE 802.15.6-Based CSMA/CA Protocol. *Journal of Medical Systems*, **36**, 3875-3891. <http://dx.doi.org/10.1007/s10916-012-9860-0>
- [5] Ullah, S. and Henry Higgin, H., Braem, B., Latre, B., Blondia, C., Moerman, I., Saleem, S., Rahman, Z. and Kwak, K.S. (2012) A Comprehensive Survey of Wireless Body Area Networks on PHY, MAC, and Network Layers Solutions. *Journal of Medical Systems*, **36**, 1065-1094. <http://dx.doi.org/10.1007/s10916-010-9571-3>
- [6] Jung, B.H., Akbar, R.U. and Sung, D.K. (2012) Throughput, Energy Consumption, and Energy Efficiency of IEEE 802.15.6 Body Area Network (BAN) MAC Protocol. *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*.
- [7] Rezvani, S. and Ghorashi, A. (2012) A Novel WBAN MAC Protocol With Improved Energy Consumption and Data Rate. *KSII Transactions on Internet and Information System*, **6**, 2302-2322.
- [8] Marinkovi, S.J., Popovici, E.M., Spagnol, C., Faul, S. and Marnane, W.P. (2009) Energy-Efficient Low Duty Cycle MAC Protocol for Wireless Body Area Networks. *IEEE Transactions on Information Technology in Biomedicine*, **13**, 915-925. <http://dx.doi.org/10.1109/TITB.2009.2033591>
- [9] Ullah, S., An, X. and Kwak, K. (2009) Towards Power Efficient MAC Protocol for In-Body and On-Body Device Networks. *Agent and Multi-Agent System: Technologies and Application*, **5559**, 335-345.
- [10] Zhen, B., Li, H.B. and Kohno, R. (2008) IEEE Body Area Networks and Medical Implant Communications. *Proceedings of the ICST 3rd International Conference on Body Area Networks*, Tempe.
- [11] Rashwand, S., Mistic, J. and Khazaei, H. (2011) IEEE 802.15.6 under Saturation: Some Problems to be Expected. *Journal of Communications and Networks*, **13**, 142-148. <http://dx.doi.org/10.1109/JCN.2011.6157413>
- [12] Rashwand, S. and Mistic, J. (2011) Performance Evaluation of IEEE 802.15.6 under Non-Saturation Condition. *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, Kathmandu.
- [13] Li, C.L., Geng, X., Yuan, J. and Sun, T. (2013) Performance Analysis of IEEE 802.15.6 MAC Protocol in Beacon Mode with Superframes. *KSII Transaction on Internet and Information Systems*, **7**, 1108-1130. <http://dx.doi.org/10.3837/tiis.2013.05.010>
- [14] Rashwand, S. and Mistic, J. (2012) Effects of Access Phases Lengths on Performance of IEEE 802.15.6 CSMA/CA. *Journal of Computer Networks*, **56**, 2832-2846. <http://dx.doi.org/10.1016/j.comnet.2012.04.023>
- [15] Bianchi, G. (2000) Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal on Selected Areas in Communications*, **18**, 535-547. <http://dx.doi.org/10.1109/49.840210>

Performance Study of Locality and Its Impact on Peer-to-Peer Systems

Mohammed Younus Talha, Rabah W. Aldhaheeri, Mohammad H. Awedh

Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia
Email: y Mohammed0003@stu.kau.edu.sa, r Aldhaheeri@kau.edu.sa, m Hawedh@kau.edu.sa

Received 15 January 2015; accepted 30 January 2015; published 2 February 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents the measurement study of locality-aware Peer-to-Peer solutions on Internet Autonomous System (AS) topology by reducing AS hop count and increase nearby source nodes in P2P applications. We evaluate the performance of topology-aware BT system called TopBT with BitTorrent (BT) by constructing AS graph and measure the hops between nodes to observe the impact of quality of service in P2P applications.

Keywords

Peer-to-Peer, BitTorrent, TopBT, Locality, Autonomous System

1. Introduction

Peer-to-Peer (P2P) is a distributed computing model which aims to share resources whose concept is not completely new. However, P2P systems are natural evolution in decentralized system architecture [1] in which peer is a node that can act as a client and server simultaneously in dynamic environment. Nodes can join or leave the system freely and also exchange resources directly without the help of a third party server. Popular P2P systems generate massive amount of traffic over the internet and it has been reported that 65% - 70% Internet backbone is P2P traffic. Furthermore, it may be estimated that 50% - 65% of download traffic and 75% - 90% of upload traffic is generated by P2P traffic access communities [2]. P2P networks can be classified according to their functionalities into three main classes such as file sharing, video streaming and VoIP. File sharing P2P applications like BitTorrent (BT), TopBT are the most popular among the three classes whereas video streaming classes applications are PPLive, PPStream and Voip application is Skype.

Zatto [3] was introduced as a localized P2P live streaming system and Skype [4] was modified to implement locality in super peers selections. Finally, Top-BT [5] was introduced as localized version of the BitTorrent software which is developed by OHIO State University R&D Dept., that actively discovers its network proximi-

ties to its connected peers, this unique feature separates it from Bit torrent.

It also improves the peers transmission rate of network for a faster download, reduces topology un-awareness due to unnecessary traffic and maintains faster download speed compared to other clients [6]. BitTorrent which is a well-known non-localized file sharing software in P2P networking used in common for transferring large file in a vast community environment with an exceptional download speed [7].

The popularity of P2P applications had massive traffic load that revealed doubts about the ability of internet service provider (ISP) that carried P2P traffic [8] and to sustain their cost of transit traffic. Due to these reasons and others inspired research to replace P2P random algorithms with locality-awareness algorithms where locality is a distance measurement method that can be utilized to express locality awareness.

Peer-to-peer (P2P) locality has recently raised lots of interest locally as its written content distribution dramatically raises the traffic within the inter-ISP links, in order to solve this problem the idea to keep a fraction in the P2P site visitors local to help each ISP has been introduced a couple of years ago. Several fundamental issues on locality are being explored such as measuring the content distribution and knowing the harmful effect of locality which intensify the demand of the content file that is shared on the network. P2P applications and ISPs have different lanes of business models that attempt to attract more users by increasing quality of service (QoS).

The fact that allowed P2P application developers to consider underlying networks as free resources and on the other hand ISP's attempting to drag down their inter/intra-domain traffic to increase their profits [9]. This business model authorized ISP's consider P2P applications as a harmful services and thus started to domesticate them by blocking their traffic with the help of shaping devices [5] and on the other side P2P applications counter-strike by encrypting their traffic using port hopping that leads to endless chasing.

However to tackle ISP issues, Autonomous system (AS) hops can be utilized to harvest AS-level topology information and closely relate the AS-based ISP pricing model. Locality awareness algorithm implementation in P2P application was widely studied in the past years [10]-[12]. Each network on the Internet is recognized by a unique identifier known as Autonomous system number (ASN) which owns a set or a block of Internet Protocol (IP) addresses that have been assigned to it, in order to prevent traffic from propagation, content should be exchanged with other IP addresses in the same AS.

Sniffing is one of the most effective techniques in attacking a wireless network. Sniffer [13] is a program that eavesdrops on the network traffic by grabbing information that travels over a network and the Source for many network-based attacks is passive sniffing. Passive sniffing involves employing a sniffer to be able to monitor these kinds of incoming packets which uses a feature connected with network greeting cards called promiscuous mode. In this mode a network card will pass all packets on the operating structure, rather than those Unicast as well as broadcast towards host [13].

World's prime network protocol analyzer named Wireshark [14] enables to capture and interactively browse the traffic flowing on a computer network. This software is customary across many industries and educational institutions. Wireshark uses a packet capture in short Pcap, an application programming language to capture packets, so it can only capture the packets on the types of networks that Pcap supports. Yi Cui, *et al.*, [15] proposes locality awareness in bit torrent like P2P applications which proposes an optimal solution with minimum AS hop count distribution structure and also describes that seeding cannot improve standard bit torrent download time but can improve its locality policies significantly.

The paper is organized as follows. Section 2 depicts the methodology of achieving goals of our study. Results analysis and performance of Locality and TopBT is studied and discussed in Section 3. Finally, Section 4 gives conclusion and possible future work to improve the quality of service in P2P applications.

2. Methodology and Data Collection

Our methodology of collecting data is to download Torrent files using two different file sharing applications such as Bit Torrent and Top-BT, which were operated in two separate computers. The download time of both Torrent clients was calculated and recorded simultaneously. Wireshark captures and save data packets of both Torrent clients. Then, a utility software was used to extract source and destination IP addresses from Wireshark captured files.

AWK tool [16] was used to delete the duplicate of the source and destination IP addresses. Cymru tool [17] had been utilized by which IP addresses were converted into Autonomous System Numbers (ASN). Java code was developed to find the AS paths from source IP address to destination IP address. By extracting these paths,

we then compared the paths generated by the BitTorrent and Top-BT applications. This procedure had been applied on different file formats (Audio, Video, Application files, etc.). Hence the whole data was collected at particular geographical location.

During the downloading of some files we came across a huge download time duration by which a user may lose interest in downloading that particular file. Investigation of the scenario was done to show the impact of locality on the quality of experience. Our calculation was based on the Autonomous System Number and after gathering these numbers, AS paths has been our metric to measure locality.

The following steps were taken to achieve our goal. First we reviewed the concepts of Peer to Peer network (P2P) and Locality revision on the P2P applications. Then, Wireshark and AWK software tools were used for data collection. Finally, Java program and Cymru software tool were employed for data analysis part by which IP addresses were extracted and has been converted to Autonomous System Numbers (ASN).

Two well-known P2P file sharing systems were utilized, namely, Bit Torrent and Top-BT. Software programming and simulation tools such as Java, AWK, Cymru were adapted to map IP's into AS numbers. MS-Excel has been used to show AS paths as final output by which the paths between Bit-torrent and TopBT are compared to measure their QoS and Locality in a P2P network.

3. Results

Locality awareness has emerged as the anchor to tackle the unwanted traffic issue where locality awareness algorithms allow peers to measure their distances from other nodes and utilizes this knowledge in selecting near content sources. To implement this algorithm, many issues must be tackled. For example, how to measure distances? How to find location? How to define near nodes and far nodes?

Collecting underlying network measurements and utilizing this information is the way to answer the previous questions. Peer should have the ability to measure the AS hop count path to reach different peers and must have the ability to map IP addresses into their AS numbers that can be able to measure delay, bandwidth and loss in the path. Finally, they should have algorithms that utilize this information (Locality algorithm). The main objective of locality awareness studies is to construct a P2P system that satisfies the requirement of ISPs by reducing the hops count and increase the number of local source nodes in one way and on the other end there shouldn't be impact on quality of experience in P2P networks.

Our results show that the average AS hops count path between neighbors in TopBT platform is shorter than the distances between neighbors in BT network. In addition, we have observed from our results that locality awareness implemented in TopBT has impact in reducing the intra-domain traffic that passes between AS's. Unfortunately, the implementation of locality awareness algorithm may reduce the performance, Quality of Service (QoS) and Quality of Experience (QoE), of P2P networks if the required content is unpopular.

In other words, the popularity of file in P2P file sharing network may affect implementation of locality awareness algorithm which means that the locality awareness algorithm requires a popularity of files to increase the performance of P2P applications or it will decrease its normal performance. In our measurement study we have obtained our results for Autonomous System paths of sources and destinations on Inter Autonomous System level routing.

In **Figure 1** we have evaluated average Autonomous System paths of Audio files in which TopBT has better download rate than BitTorrent. **Figure 2** and **Figure 3** shows the AS source path comparison of Video files and Application files respectively whereas the Video files that contain large data size results in such case TopBT has performed good as shown in **Figure 2**. Finally, we note that **Figure 4** show Document source average AS path files.

In P2P networks, nodes act as client and server simultaneously. **Figure 5** shows average AS paths of Audio destination files, in **Figure 6** and **Figure 7** average AS path destination files of Video and Application are shown in which the performance of BitTorrent is slight better than TopBT. However TopBT had good performance overall, whereas **Figure 8** shows the average AS hops path of destinations of Document files to compare the performance of TopBT with BitTorrent.

By observing these average AS paths of source files and destinationfiles figures respectively we can list out our findings, At first we noticed average AS hops paths of TopBT is shorter in most of the cases. However, in some cases this path is longer than BitTorrent. The reason is that the downloaded files in these cases are not popular, which means that there are no localized nodes near to download the file from. In this case TopBT attempts

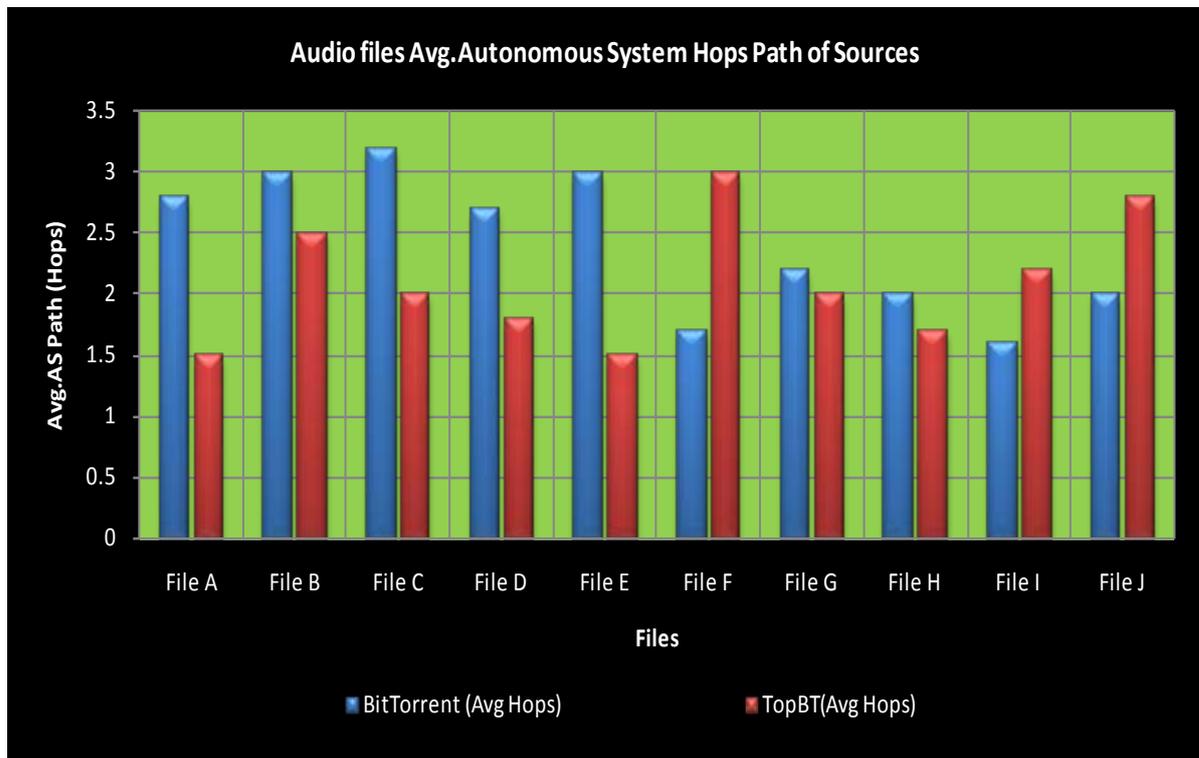


Figure 1. Source autonomous system hops path for audio files.

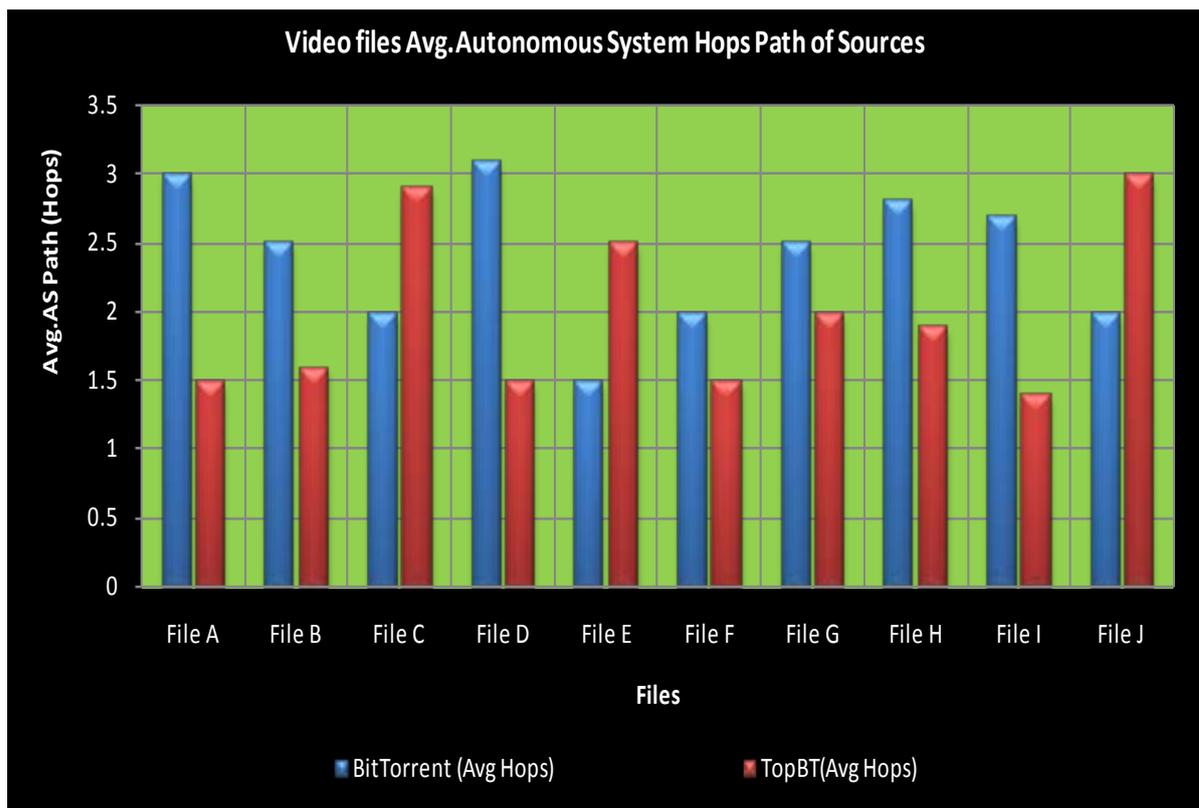


Figure 2. Source autonomous system hops path for Video files.

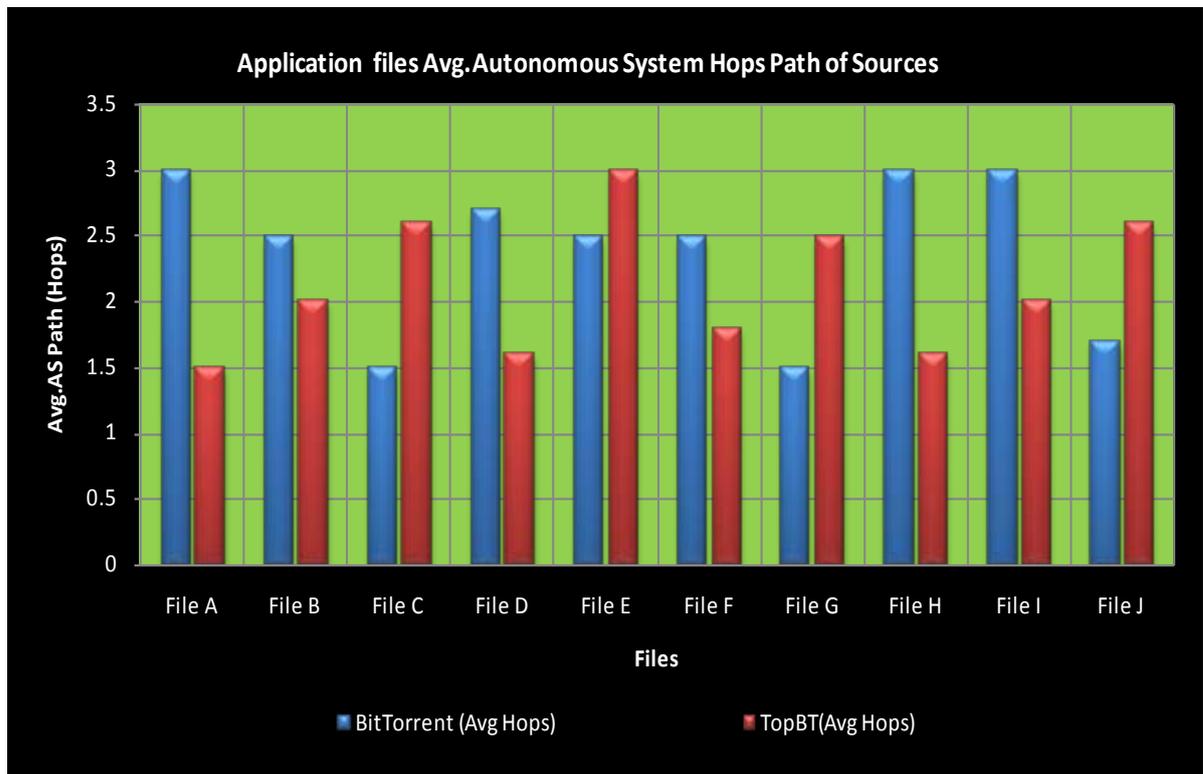


Figure 3. Source autonomous system hops path for application files.

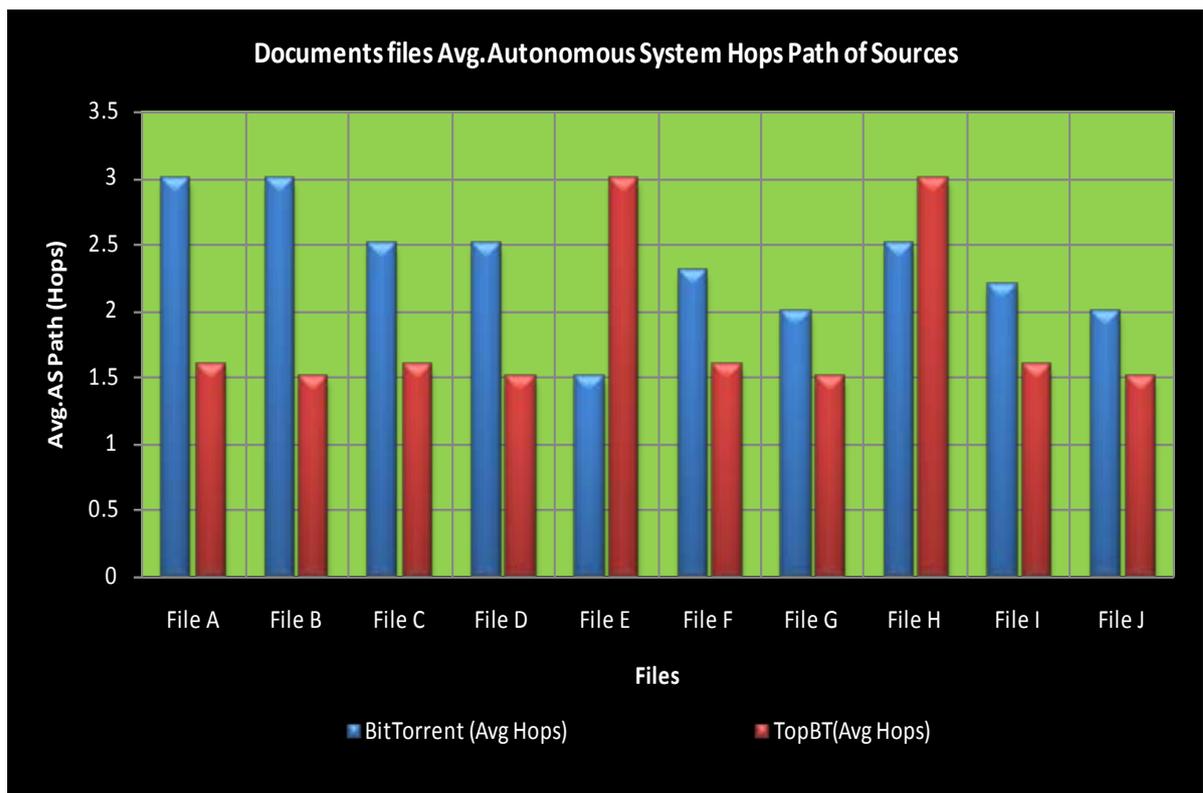


Figure 4. Source autonomous system hops path for documents files.

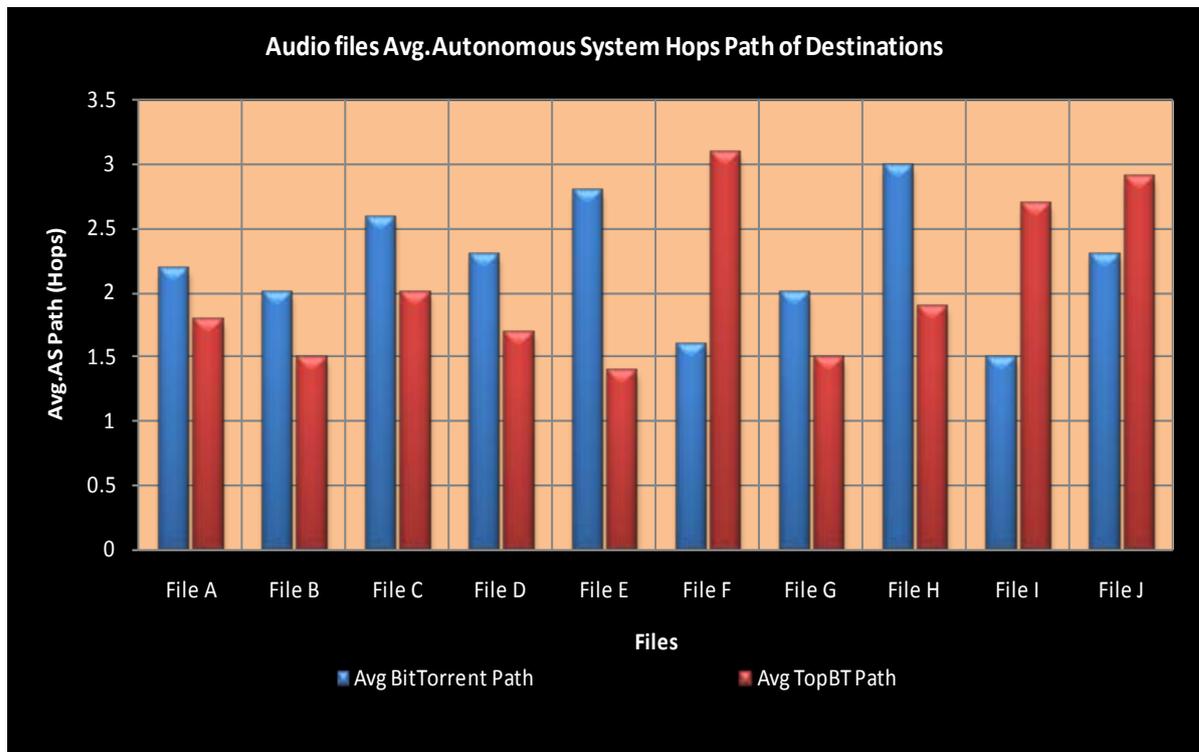


Figure 5. Destination autonomous system hops path for audio files.

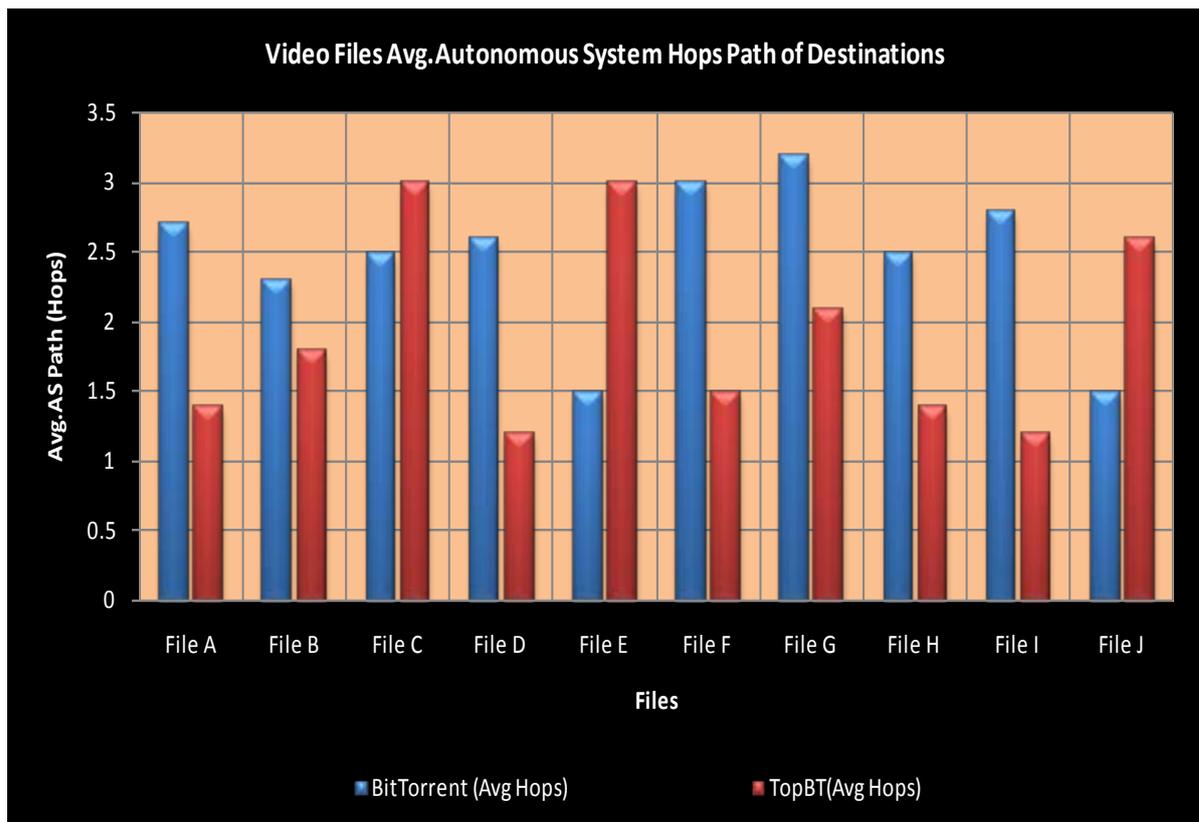


Figure 6. Destination autonomous system hops path for video files.

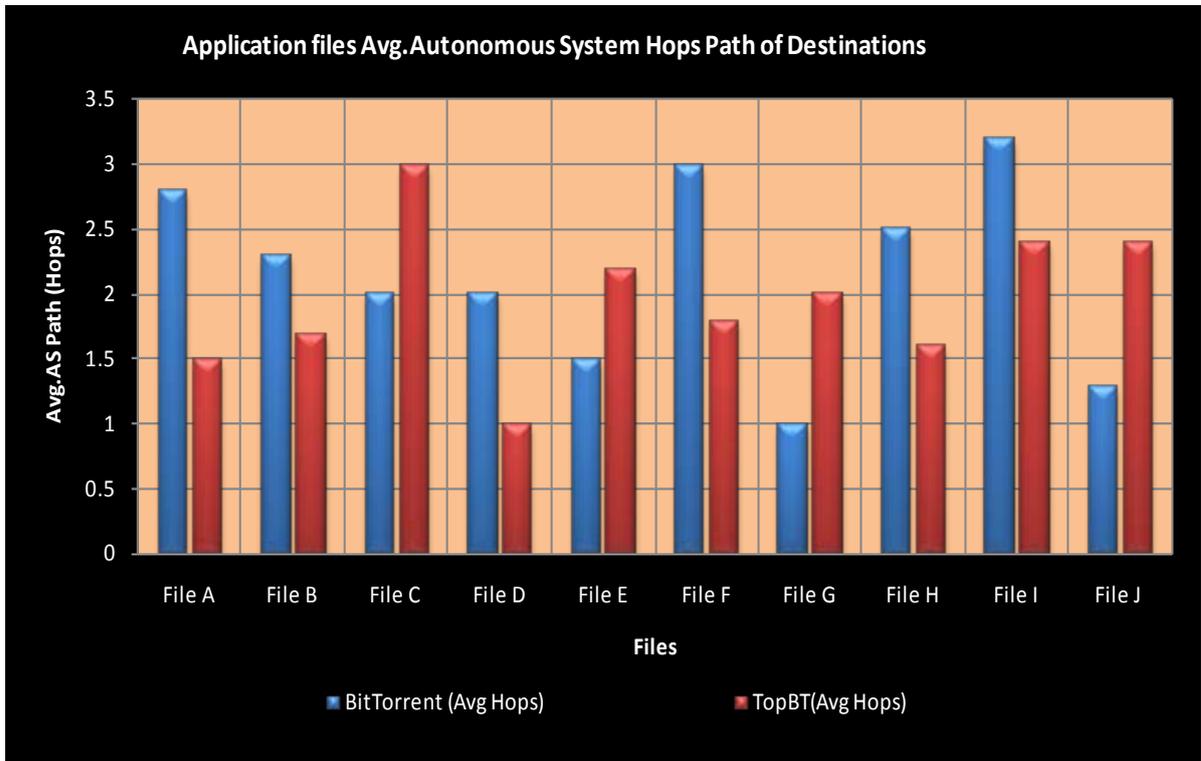


Figure 7. Destination autonomous system hops path for application files.

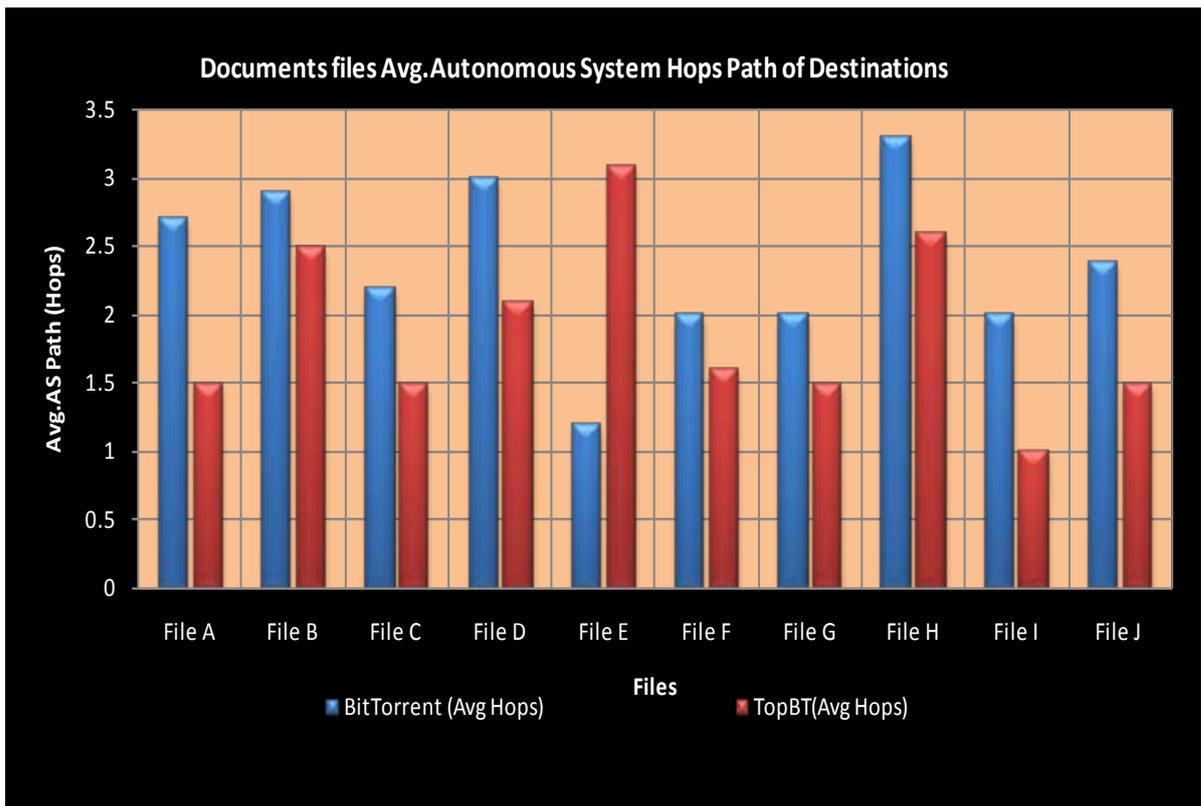


Figure 8. Destination autonomous system hops path for documents files.

to download the files from faster nodes with highest upload bandwidth and hence this fact has increased the path for such scenarios. We can observe that the download time has reduced in many scenarios for TopBT, this shows that locality can also help in improving the Quality of Experience. Unfortunately, this is not the case for all files. TopBT attempts to reduce paths' length first which can affect the download time by downloading the file from closer procrastinate nodes.

4. Conclusions & Future Work

In this work, we conducted a measurement study to investigate the advantages and drawbacks of implementing locality awareness algorithm in P2P networks and examined the locality awareness algorithm in BitTorrent and TopBT. We have compared the performance of TopBT with BitTorrent and utilized Wireshark tool to collect information from P2P network. In addition, an AS graph has been constructed to implement a shortest path algorithm to measure AS hops count between nodes in which collected peers from BitTorrent have been used as input to measure their destinations.

In future work, we can use other P2P applications and compare their results with each other and also investigate this measurement study in different locations and diverse ISP internet connections.

P2P model will remain dominating in coming years as we believe that research and development will continue to adapt P2P overlays which are more suitable for current internet infrastructure. P2P systems evolution will provide insights into the development of other large-scale distributed systems.

References

- [1] Quang, H.V., Minhai, L. and Beng, C.O. (2010) Peer to Peer Computing Principles and Applications. New York City. <http://dx.doi.org/10.1007/978-3-642-03514-2>
- [2] Masoud, M.Z., Hei, X.J. and Cheng, W.Q. (2012) Constructing a Locality-Aware ISP-Friendly Peer-to-Peer Live Streaming Architecture. *Proceeding of International Conference on Information Science and Technology (ICIST)*, Hubei, 23-25 March 2012, 368-376. <http://dx.doi.org/10.1109/ICIST.2012.6221670>
- [3] Alhaisoni, M., Liotta, A. and Ghanbari, M. (2009) Performance Analysis and Evaluation of P2PTV Streaming Behavior. *International Symposium on Computers and Communications (ISCC)*, Sousse, 89-94. <http://dx.doi.org/10.1109/ISCC.2009.5202402>
- [4] Zhang, D.Y., Zheng, C., Zhang, H.L. and Yu, H.L. (2010) Identification and Analysis of Skype Peer-to-Peer Traffic. *International Conference on Internet and Web Applications and Services (ICIW)*, Barcelona, 200-206. <http://dx.doi.org/10.1109/ICIW.2010.36>
- [5] Aggarwal, V., Akonjang, O. and Feldmann, A. (2008) Improving User and ISP Experience through ISP-Aided P2P Locality. *Proceedings of INFOCOM Workshops*, Phoenix, 1-6. <http://dx.doi.org/10.1109/INFOCOM.2008.4544640>
- [6] Ren, S., Tan, E., Luo, T., Guo, L., Chen, S. and Zhang, X. (2010) TopBT: A Topology-Aware and Infrastructure-Independent BitTorrent Client. *Proceedings of INFOCOM*, San Diego, 1-9. <http://dx.doi.org/10.1109/INFOCOM.2010.5461969>
- [7] Xia, R.L. and Muppala, J.K. (2010) A Survey of BitTorrent Performance. *International Journal of Communications Surveys & Tutorials*, **12**, 140-158. <http://dx.doi.org/10.1109/SURV.2010.021110.00036>
- [8] Frasn, M., Klampfer, S. and Cucej, Z. (2008) Impact of P2P Traffic on IP Communication Network Performances. *Proceeding of International Conference on Systems, Signals and Image Processing*, Bratislava, 205-208. <http://dx.doi.org/10.1109/IWSSIP.2008.4604403>
- [9] Blond, S.L., Legout, A. and Dabbous, W. (2004) Pushing BitTorrent Locality to the Limit. *International Journal of Computer and Telecommunications Networking*, **55**, 541-557. <http://dx.doi.org/10.1016/j.comnet.2010.09.014>
- [10] Coffins, D.R. and Bustamante, F.E. (2008) Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in Peer-to-Peer Systems. *Proceeding of ACM SIGCOMM Conference on Data Communication*, 363-374. <http://dx.doi.org/10.1145/1402958.1403000>
- [11] Lewis, P.R., *et al.* (2011) A Survey of Self-Awareness and Its Application in Computing Systems. *Proceeding of International Conference on Self-Adaptive and Self-Organizing Systems Workshops (ICSASOW)*, Ann Arbor, 102-107. <http://dx.doi.org/10.1109/SASOW.2011.25>
- [12] Agarwal, V., Feldmann, A. and Scheideler, C. (2007) Can ISPs and P2P Users Cooperate for Improved Performance. *ACM SIGCOMM Computer Communication Review Journal*, **37**, 29-40.
- [13] Chomsiri, T. (2008) Sniffing Packets on LAN without ARP Spoofing. *Proceeding of International Conference on Convergence and Hybrid Information Technology (ICCIT)*, **2**, 472-477. <http://dx.doi.org/10.1109/ICCIT.2008.318>

- [14] Wang, S.Q., Xu, D.S. and Yan, S.L. (2010) Analysis and Application of Wireshark in TCP/IP Protocol Teaching. *Proceeding of International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT)*, **2**, 269-272. <http://dx.doi.org/10.1109/EDT.2010.5496372>
- [15] Liu, B., Cui, Y., Lu, Y.S. and Xue, Y. (2009) Locality-Awareness in BitTorrent-Like P2P Applications. *IEEE Transactions on Multimedia—Special Section on Communities and Media Computing Journal*, **11**, 361-371. <http://dx.doi.org/10.1109/TMM.2009.2012911>
- [16] Aho, A.V., Kernighan, B.W. and Weinberger, P.J. (1988) *The AWK Programming Language*. New York.
- [17] Cymru. <http://www.team-cymru.org/Services/ip-to-asn.html>

Improving Queuing System Throughput Using Distributed Mean Value Analysis to Control Network Congestion

Faisal Shahzad¹, Muhammad Faheem Mushtaq¹, Saleem Ullah^{1*}, M. Abubakar Siddique², Shahzada Khurram¹, Najia Saher¹

¹Department of Computer Science & IT, The Islamia University of Bahawalpur, Bahawalpur, Pakistan

²College of Computer Science, Chongqing University, Chongqing, China

Email: faisalsd@gmail.com, faheem.mushtaq88@gmail.com, saleemullah@iub.edu.pk, abubakar.ahmadani@gmail.com, khurram@iub.edu.pk, najia@iub.edu.pk

Received 12 June 2014; accepted 30 January 2015; published 2 February 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we have used the distributed mean value analysis (DMVA) technique with the help of random observe property (ROP) and palm probabilities to improve the network queuing system throughput. In such networks, where finding the complete communication path from source to destination, especially when these nodes are not in the same region while sending data between two nodes. So, an algorithm is developed for single and multi-server centers which give more interesting and successful results. The network is designed by a closed queuing network model and we will use mean value analysis to determine the network throughput (β) for its different values. For certain chosen values of parameters involved in this model, we found that the maximum network throughput for $\beta \geq 0.7$ remains consistent in a single server case, while in multi-server case for $\beta \geq 0.5$ throughput surpass the Marko chain queuing system.

Keywords

Network Congestion, Throughput, Queuing System, Distributed Mean Value Analysis

1. Introduction

Networks where a communication path between nodes doesn't exist refer to delay tolerant networks [1] [2]; they

*Corresponding author.

communicate either by already defined routes or through other nodes. Problem arises when network is distributed and portioned in to several areas due to the high mobility or when the network extends over long distances and low density nodes. The traditional approach [3] for the queuing system was to design a system of balance equations for the joint property of distributed vector value state. The traditional approaches to the system of Markovian [4] queuing systems were to formulate a system of algebraic equations for the joint probability distributed system vector valued state, which was the key step introduced by Jackson [5], that for a certain types of networks the solutions of the balance equation is in the form of simple product terms. All remained to be normalized numerically to form the proper probability distribution In case of networks with congested routing chains, this normalization turned out to be limited and degrades the system efficiency as well.

Therefore, in practical these distribution contains an extra detail, such as mean queue size, mean waiting time and throughput is needed. The framework of conventional algorithm shows that these properties can be obtained by normalizing constants. The proposed algorithm given in this paper correlates directly with the required statistics. Its complexity is asymptotically is almost equal to the already defined algorithms, but the implementation of program is very simple.

Choosing the right queuing discipline and the adequate queue length (how long a packet resides in a queue) may be a difficult task, especially if your network is a unique one with different traffic patterns. Monitoring of the network determines which queuing discipline is adequate for the network. It is also important to select a queuing length that is suitable for your environment. Configuring a queue length that is too shallow could easily transmit traffic into the network too fast for the network to accept, which could result in discarded packets. If the queue length is too long, you could introduce an unacceptable amount of latency and Round-Trip Time (RTT) jitter. Program sessions would not work and end-to-end transport protocols (TCP) would time out or not work.

Because queue management is one of the fundamental techniques in differentiating traffic and supporting QoS functionality, choosing the correct implementation can contribute to your network operating optimally.

2. Mean Value Evaluation

Included in the Quality of Service Internetwork architecture is a discipline sometimes called queue management. Queuing is a technique used in internetwork devices such as routers or switches during periods of congestion. Packets are held in the queues for subsequent processing. After being processed by the router, the packets are then sent to their destination based on priority.

In the queuing network, the traditional approach to find the solution is using characteristics of a continuous time markov chain to formulate a system of balance equation for the joint probability distribution of the system state. The solution of the balance equations, for certain classes of networks such as Jackson networks and Gordon-Newell networks is in the form of a product of simple terms, see [6]. In general, the joint probability is not so simple and sometime it is inefficient to pursue. If we only interested in the average performance measure such as average waiting time, average response time or network throughput, we do not need the steady-state probabilities of the queue length distributions. In this stage, we will use an approach so called mean value analysis. The algorithm for this approach works directly to the desired statistics and has been developed and applied to the analysis of queuing networks, for example see [7]. The mean value analysis is based on the probability distribution of a job at the moment it switches from one queue at a server to another and the steady state probability distribution of jobs with one job less. This relation is known as arrival theorem for closed queuing networks. In a closed queuing network, the bottleneck is the queue with the highest service demand per passage.

Using the arrival theorem, if a job move from queue i to queue j in a closed queuing networks with K jobs in it, will find on average $E[N_i(K-1)]$ jobs. With this result and assuming that a job is served in a first come first served basis, a relation between the average performance measure in this network with K jobs and $K-1$ job can be performed recursively.

3. Proposed Mechanism

Mean value analysis depends on the mean queue size and mean waiting time. This equation applied to each routing chain and separately to each service center will furnish the set of equations which will easily solved numerically. The proposed algorithm is simple and avoid overflow, underflow actions which may arise with traditional algorithms. All mean values in the algorithm are calculated in a parallel manner. Thus memory requirement is higher than the previous ones, but new mechanism is relatively faster in multi-server scenarios.

We have considered the closed multi-chain queuing system which has the product form solution. Suppose C is a routing chain and S is a service center. Each chain contains a fixed number of customers who processed through subset of services using Markov chain technique, while service providers adopt one of the following mechanisms.

- 1) FIFO: customers are serviced in order of arrival, and multi-servers can be used.
- 2) Priority Queuing: customers are serviced according to the traffic categorization.
- 3) WFQ (Weighted Fair Queuing): gives low-volume traffic flows preferential treatment and allows high-volume traffic flows to obtain equity in the remaining amount of queuing capacity. WFQ tries to sort and interleave traffic by flow and then queues the traffic according to the volume of traffic in the flow.
- 4) PS: Customers are served in parallel by a single server.
- 5) LCFSRP: customers are served in reverse order of arrival by a single server, (Last come first served preemptive resume).

Now we assume that all the servers have constant service rate using multiple FCFS service centers starting with the following consequences, which relates mean waiting time $W(K)$ to the mean queue size of the system $W(K - Er)$ with one customer less in the chain r making the following equation,

$$T_{r,l} = \sigma_{r,l} \{1 + n_l (K - E_r)\} \quad (1)$$

$T_{r,l}$ is the equilibrium mean waiting time of chain r at service center l $\sigma_{r,l}$ is the mean value of the service demand, n_l represent equilibrium mean queue size, K is size of chain r , and E_r is the R -dimensional unit vector. From the definition it is clear that

$$T_{r,l} = \sigma_{r,l} \quad (2)$$

K_r is the average number of customers in chain between successive visits, then the mean number of visits (θ) and its waiting time per visit with service center visited by chain $S(r)$ is described as

$$\alpha_r = k_r \left[\frac{1}{\sum_{l \in S(r)} (\theta_{r,l} T_{r,l})} \right] \quad (3)$$

where α_r is the throughput of chain r , considering service center $l(r)$ i.e. $\alpha_r = \alpha_{r,l(r)}$ then for multiple service stations each one yields the following relation,

$$N_{r,l} = \alpha_{r,l} T_{r,l} = \alpha_r \theta_{r,l} T_{r,l} \quad (4)$$

The above equations applicable for recursive analysis of mean queuing size, meat waiting time and system throughput. The initial point can be set as, $N_{r,l}(0) = 0$ For all $r = 1, \dots, R$ and $l = 1, \dots, L$.

To make the substitution in an algorithmic form, we have $T'_{r,l} = \theta_{r,l} T_{r,l}$ and $N'_l = 1 + N_l$.

3.1. Mathematical Model

Our model is defined on a one-dimensional closed system consisting of M cells i.e. **Figure 1**. A closed queuing network model is justified for steady state conditions. In steady state, for a single-entry and single-exit lane, the traffic flow into the system will be equal to the traffic flow out of the system. We approximate this as a closed system where the number of vehicles remains the same. Each cell can either be empty or occupied by one vehicle. To start with, we assume vehicles of identical size. Since the system is closed, the number of vehicles remains constant, say equal to N . Thus the system density can be defined as $N/M = \rho$. Hereon, this model will be referred to as the Path Cell Network model. A vehicle moves from the first to the second and so on to the M^{th} cell and then back to the first cell. It is apparent that the system has attributes of a queuing system with FIFO discipline. In the past, the general modeling of traffic using Queuing Theory has been macroscopic, but here instead of treating the whole closed link as a single queue, we consider it as a network of queues. The Path Cell Network model at first appears to be a cumbersome one as each cell has limited space and, therefore, each queue in the network limited buffer space. But, our task can be made much easier by our definition of the servers. The exact working of the model is as follows:

Being a single lane model each vehicle moves to the next cell if empty or waits, and then moves when the vehicle ahead vacates the cell. Thus there can only be two configurations for a particular vehicle: either the cell

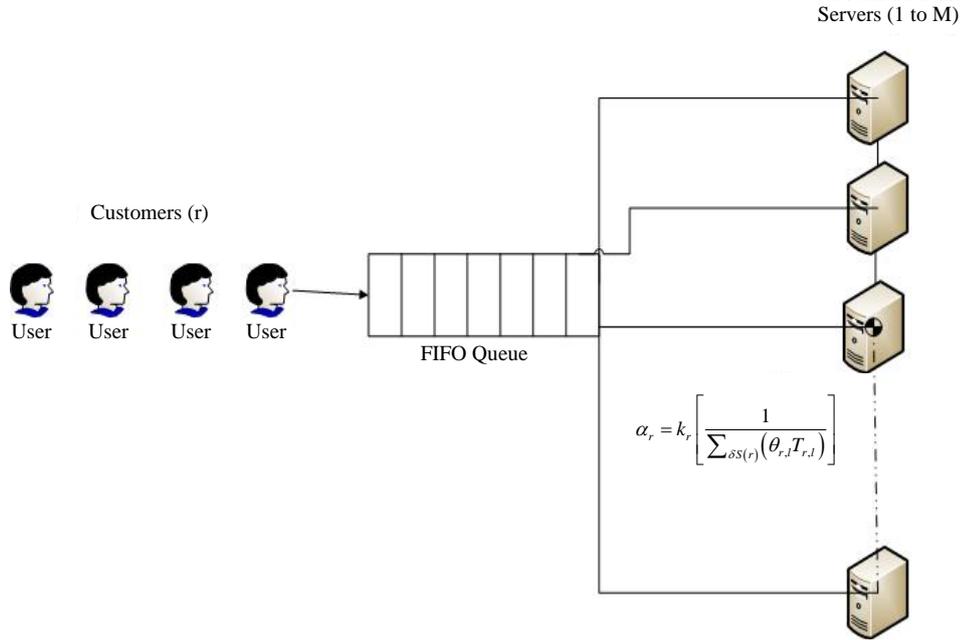


Figure 1. Queuing model.

ahead is empty or occupied. We say a vehicle is in service when the cell ahead is empty and “waiting” if it is occupied.

```

if (cell = occupied)
    wait
else
    In_service
    
```

In effect, each empty cell acts as a server, and at any point in time there are always $M - N$ servers in the system that keeps changing their positions. These dynamic cells act as servers to $M - N$ queues in the system that together form a closed network. The number of waiting units in each queue can be counted as the total number of vehicles between the empty cell and the one empty behind it. Thus if there are two consecutive empty cells, both act as servers with one of them having zero queue size. Service in each queue is assumed to be exponential, and for the basic Path Cell Network model, assuming identical vehicles, the service rate of each vehicle is also taken to be the same. As the service is exponential, from the Poisson-in-Poisson-out property inter-arrival times at each queue are also exponential.

The model we claim can be mapped onto a cyclic Jackson network with $M - N$ cells and N customers. Thus, effectively, a path segment with very limited buffer space at each queue is mapped onto a well-known cyclic queuing network with buffer space of size N .

The results of a cyclic Jackson network are well known. A state is indicated by

$$k_1 k_2 k_3 \dots k_n, \text{ where } \sum k_i = N$$

and k_i indicates the number of units at each stage of the closed queuing network. The probability of being in a state $k_1 k_2 k_3 \dots k_n$ is written as $p(k_1 k_2 k_3 \dots k_n)$. Transitions between states occur when a unit enters or leaves a stage. Service rate μ_i at each stage is allowed to be dependent upon the number of units i in the stage. Then

$$p(k_1 k_2 k_3 \dots k_n) = p(N, 0, 0, 0) \frac{\mu_1^{N-k_1}}{\mu_2^{k_2} \mu_3^{k_3} \dots \mu_n^{k_n}} \tag{5}$$

while, $\sum (k_1 k_2 k_3 \dots k_n) = 1$

In this case we have consider the service rate and probability at each node and stage is same and equal to the inverse of the number of ways of selecting N out of $N+k-1$ places as customers and remaining $k-1$ places being the partitions, it is calculated as

$$p = \frac{N!(k-1)}{k+N-1},$$

here throughput α_j of the network can be calculated as

$$\alpha_j = \mu p(k_1 > 0) = \mu(1 - p(k_1 = 0)) \quad (6)$$

where $p(n_1=0)/p$ is simply the number of ways of selecting N out of the last $N+k-2$ places as customers, the first place being a partition

$$\alpha_j = \mu \frac{N}{N+k-1} \quad (7)$$

For the corresponding network number of queues $k = M - N$, throughput β_r of the network is obtained by scaling α_j using the following relation

$$M_{\beta_r} = (M - N)/\alpha_j \quad (8)$$

After scaling, throughput becomes

$$\beta_r = \mu \frac{N(M - N)}{(M - 1)M} \quad (9)$$

For $N, M \gg 1$, $\beta_r = \mu p(1 - p)$

The algorithm starts with an empty network (zero customers), then increases the number of customers by 1 until it reaches the desired number of customers of chain r .

The average waiting time in this closed queuing network and the average response time per visit are given by the following formulas:

$$E[w_i(k)] = E[N_i(k-1)]E(s_i) \quad (10)$$

where, s_i is the service time, and w_i is waiting time. To obtain the average response time per passage, the above equation can be obtained by each time visit ratio as

$$E[P_i(k)] = E[N_i(k-1)+1]E(s_i)v_i \quad (11)$$

where, p_i is response time and v_i is each time visit. The expected number of jobs in queue I is given by little's formula [6] as.

$$E[N_i(k)] = \beta(k)E[p_i(k)] \quad (12)$$

3.2. Case Studies

Now we have to implement our model in single and multi-server scenarios to calculate throughput and mean waiting time.

a) SINGLE SERVER CASE

Initialize $N'_l[0] = 0$ for all $l = 1, 2, 3, \dots, L$

for $(i_1 = 0, \dots, K_1)$, for $(i_2 = 0, \dots, K_2)$, and for $(i_R = 0, \dots, K_R)$

If at the service centers customers are delayed independently (D) of other customers, then we have the fol-

lowing steps,

$$\left. \begin{array}{l} \text{if (Service_Center == } D) \\ T'_{r,l} = Y_{r,l} \delta(i_r) \\ \text{else} \\ T'_{r,l} = Y_{r,l} N'_l(i - e_r) \end{array} \right\} \forall r = 1, 2, \dots, R \quad (13)$$

Then we have a little's equation for chains and service centers as,

$$\alpha_r = \frac{l_r}{\sum_{l \in \mathcal{S}(r)} T'_{r,l}}, \quad \forall r = 1, \dots, R \quad (14)$$

$$T'_l(I) = 1 + \sum_{r \in R(l)} \alpha_r T'_{r,l}, \quad \forall l = 1, 2, \dots, L \quad (15)$$

The operations count for this algorithm is bounded by $2RL - R$ additions and $2RL + R$ multiplication/divisions. This is the same as the convolution algorithm in its most efficient. However, Algorithm 1 completely avoids a genuine problem of the convolution algorithm, namely, that the floating point range of many computers may be easily exceeded. Scaling, as discussed in [8], may partially alleviate the problem. Yet the scaling algorithm is complex and does not always work. The authors have seen several well-posed modeling problems involving relatively large populations (e.g., >100) and type D service centers which were not solvable in the range of floating point numbers $1E \pm 75$, despite scaling. The storage requirement is of the order $LK_1K_2 \dots K_R$ as compared to $2K_1K_2 \dots K_R$ for the convolution algorithm.

We now proceed to extend the computational procedure to handle FCFS service centers with multiple constant unit rate servers. The mean value Equation (14) for such a center can be written as

$$T_{r,l} = \frac{\pi_l}{M_l} \left[1 + T_l(k - e_r) + \sum_{i=0}^{M_l-2} (M_l - 1 - i) p_l(i, k - e_r) \right] \quad (16)$$

where (π_l) is the mean service demand, which is assumed to be independent of the chain. The calculation is complicated by the marginal queue size probabilities, which we have to carry along in the recursive scheme. This can be done by means of Lemma 1, which allows calculation of $p_l(i, K)$, $i = 1, 2, \dots, M - 1$ from previously computed values. In order to keep the recursion going, we need an independent equation for $p_l(0, K)$, which is obtained from the following relations

$$\sum_{i=0}^{M_l} (M_l - i) P_l(i, k) = M_l - \tau_l \quad (17)$$

where

$$\tau_l = \pi_l \sum_{r=1}^R \alpha_{r,l} \quad (18)$$

From Equations (17) & (18) we can have the mean number of idle servers as $M_l - \tau_l$, which is just like a little's equation implemented to the set of servers.

b) MULTISERVER CASE

Step 1: Parameters Initialization

$$\begin{aligned} N_l(0) &= 1, \\ P_l(0, 0) &= 1, \\ P_l(l, 0) &= 0, \quad \forall l = 1, 2, 3, \dots, L. \end{aligned}$$

Step 2: Main Loop, same as in single server case.

Step 3: Additional Corollary for Multi-servers.

$$T'_{r,l} = \frac{p_{r,l}}{M_l} \left\{ N_l(i - k_r) + \sum_{j=0}^{M_l-2} (M_l - 1 - j) p_l(j, 1 - k_r) \right\} \quad (19)$$

For $r = 1, 2, \dots, R$ and each FCFS multi server center $l \in S(r)$, while for other service centers use the equation of step 3 in single server case.

Step 4: Little's equation for chains having α_r , same as in single server case.

Step 5: Little's equation for service centers having $N'_l(i)$, same as in single server case.

Step 6: Additional step for calculating marginal queue size under main loop for each multi FCFS service center l and $j = 1, 2, \dots, M_l - 1$.

$$P_l(j, i) = \frac{1}{J} \sum_{r \in R(l)} \alpha_r p_{r,l} p_l(J - 1, i - e_r) \quad (20)$$

$$\tau_l = \sum_{r \in R(l)} \alpha_r p_{r,l} \quad (21)$$

$$P_l(0, i) = 1 - \frac{1}{M_l} \left[\tau_l + \sum_{j=1}^{M_l-1} (M_l - j) p_l(j, i) \right] \quad (22)$$

The process evaluate per multi service center and per recursive step of the order $2(M + 1)R$ additions and $3MR + 2M$ multiplications. We observe that it grows linearly with M .

c) Queuing Theory Limitations

The assumptions of classical queuing theory may be too restrictive to be able to model real-world situations exactly. The complexity of production lines with product-specific characteristics cannot be handled with those models. Therefore specialized tools have been developed to simulate, analyze, visualize and optimize time dynamic queuing line behavior.

For example; the mathematical models often assume infinite numbers of customers, infinite queue capacity, or no bounds on inter-arrival or service times, when it is quite apparent that these bounds must exist in reality. Often, although the bounds do exist, they can be safely ignored because the differences between the real-world and theory is not statistically significant, as the probability that such boundary situations might occur is remote compared to the expected normal situation. Furthermore, several studies [9] [10] show the robustness of queuing models outside their assumptions. In other cases the theoretical solution may either prove intractable or insufficiently informative to be useful.

Alternative means of analysis have thus been devised in order to provide some insight into problems that do not fall under the scope of queuing theory, although they are often scenario-specific because they generally consist of computer simulations or analysis of experimental data.

4. Simulation Results & Discussion

The network bottleneck is the fast server. For $\beta > 0.7$ the fast server is also the network bottleneck, but when $\beta < 0.7$, the network bottleneck is the slow server.

The determination of network throughput for different values of β is calculated recursively. Every job arrives at server serve immediately (FCFS). **Figure 2** shows the plot between β and network throughput and clearly shows the difference between two schemes with consistence behavior.

Figure 3 describes value by value β performance with rival scheme. Markov Chain scheme started very confidently achieving 35 Mbps at $\beta = 0.1$ but later at $\beta = 0.5$ our proposed scheme performance increases in terms of Mbps which remain consistent. So in multi-server case for every value of $\beta \geq 0.5$ the queuing systems performs well. After deep analysis on other resulted files created after simulation, we have seen that size of packet is also increases as throughput increase which also help in improving system overall performance.

Figure 4 shows queuing system mean waiting time which increases as usual as the number of customer in chain increases. But still our proposed model is somehow better while comparing with other.

This means waiting time also one of the main objectives of my future work. To define and construct a model in which mean waiting system decreases as the number of customers increases by implementing some grid

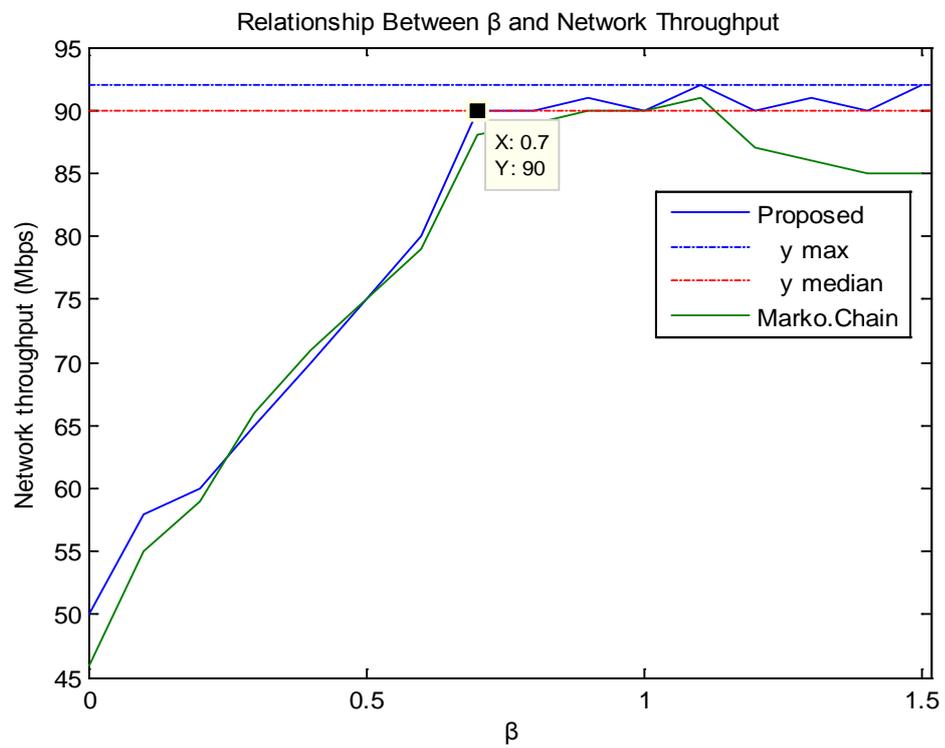


Figure 2. Throughput and consistency behavior between two schemes $\beta_r = \mu \frac{N(M-N)}{(M-1)M}$.

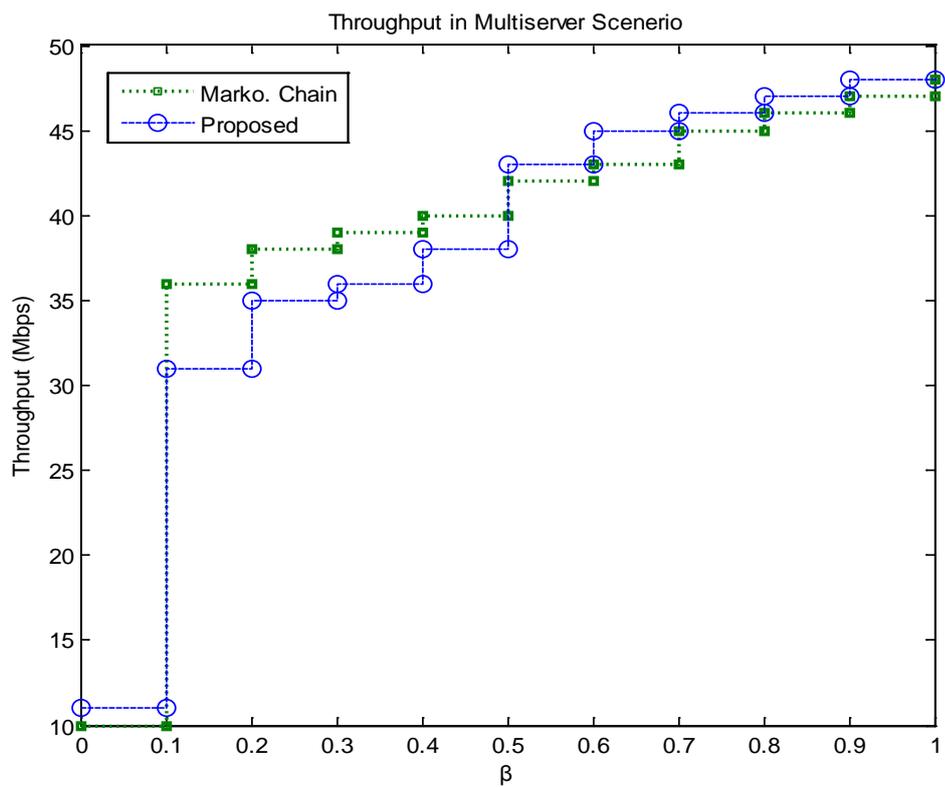


Figure 3. Throughput and consistency behavior (Multi-Server Scenerio).

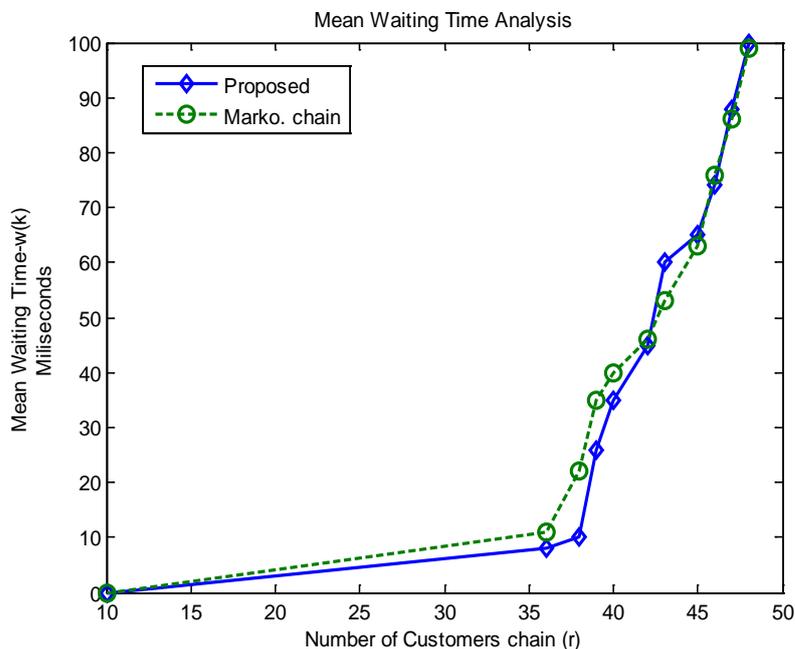


Figure 4. Queuing mean waiting time.

computing functionalities. Also developing a queuing network model for multi-hop wireless ad hoc networks keeping same objectives, used diffusion approximation to evaluate average delay and maximum achievable per-node throughput. Extend analysis to many to one case, taking deterministic routing into account.

Acknowledgements

This research work was partially supported by NSF of china Grant No. 61003247. The authors also would like to thanks the anonymous reviewers and the editors for insightful comments and suggestions.

References

- [1] Jones, E.P.C., Li, L. and Ward, P.A.S. (2005) Practical Routing in Delay-Tolerant Networks. *Proceedings of ACM SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, New York, 1-7.
- [2] Zhao, W., Ammar, M. and Zegura, E. (2005) Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network. *INFOCOM*, 1407-1418.
- [3] Bambos, N. and Michalidis, G. (2005) Queuing Networks of Random Link Topology: Stationary Dynamics of Maximal Throughput Schedules. *Queueing Systems*, **50**, 5-52. <http://dx.doi.org/10.1007/s11134-005-0858-x>
- [4] Kawanishi, K. (2005) On the Counting Process for a Class of Markovian Arrival Processes with an Application to a Queuing System. *Queueing Systems*, **49**, 93-122. <http://dx.doi.org/10.1007/s11134-005-6478-7>
- [5] Masuyama, H. and Takine, T. (2002) Analysis of an Infinite-Server Queue with Batch Markovian Arrival Streams. *Queueing Systems*, **42**, 269-296. <http://dx.doi.org/10.1023/A:1020575915095>
- [6] Armero, C. and Conesa, D. (2000) Prediction in Markovian Bulk Arrival Queues. *Queueing Systems*, **34**, 327-335. <http://dx.doi.org/10.1023/A:1019121506451>
- [7] Bertsimas, D. and Mourtzinou, G. (1997) Multiclass Queuing System in Heavy Traffic: An Asymptotic Approach Based on Distributional and Conversational Laws. *Operations Research*, **45**, 470-487. <http://dx.doi.org/10.1287/opre.45.3.470>
- [8] Reiser, M. (1977) Numerical Methods in Separable Queuing Networks. *Studies in Management SCI*, **7**, 113-142.
- [9] Chandy, K.M., Howard, J., Keller, T.W. and Towsley, D.J. (1973) Local Balance, Robustness, Poisson Departures and the Product Forms in Queuing Networks. Research Notes from Computer Science Department, University of Texas at Austin, Austin.
- [10] Knadler Jr., C.E. (1991) The Robustness of Separable Queuing Network Models. In: Nelson, B.L., David Kelton, W. and Clark, G.M., Eds., *Proceedings of the 1991 Winter Simulation Conference*, 661-665.

Hard Decision-Based PWM for MIMO-OFDM Radar

Omar Daoud

Communications and Electronics Engineering Department, Philadelphia University, Amman, Jordan
Email: odaoud@philadelphia.edu.jo

Received 15 January 2015; accepted 6 February 2015; published 9 February 2015

Copyright © 2015 by author and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

For the purpose of target localization, Multiple Input Multiple Output-Orthogonal Frequency Division Multiplexing (MIMO-OFDM) radar has been proposed. OFDM technique has been adopted in order to a simultaneous transmission and reception of a set of multiple narrowband orthogonal signals at orthogonal frequencies. Although multi-carrier systems such as OFDM support high data rate applications, they do not only require linear amplification but also they complicate the power amplifiers design and increase power consumption. This is because of high peak-to-average power ratio (PAPR). In this work, a new proposition has been made based on the Pulse Width Modulation (PWM) to enhance the MIMO-OFDM radar systems' performance. In order to check the proposed systems performance and its validity, a numerical analysis and a MATLAB simulation have been conducted. Nevertheless of the system characteristics and under same bandwidth occupancy and system's specifications, the simulation results show that this work can reduce the PAPR values clearly and show capable results over the ones in the literature.

Keywords

Multiple Input Multiple Output, Orthogonal Frequency Division Multiplexing, RADAR, Peak to Average Power Ratio, Pulse Width Modulation

1. Introduction

Many researchers have turned their attentions toward the Orthogonal Frequency Division Multiplexing (OFDM) scheme in order to provide high data rate applications under maintaining the spectral efficiency. Therefore, its clearly deployed in many broadband communication systems and protocols such as WiFi, WiMax, 4G and advanced LTE, Bluetooth-2. However, due to a high Peak-to-Average Power Ratio (PAPR), linear amplifiers suffer from low power efficiency under the utilization with multicarrier systems [1]-[5]. As a consequence, the cost of such devices; power amplifiers, mixers and analogue to digital convertor will be increased [1] [6]. OFDM

signals are easily generated and produced by applying the Fast Fourier Transform (FFT) processing block and its Inverse (IFFT) [7]. This is due to its high speed processing in performing the needed operations, such as the transformation process, filtering and correlation [8]. On the other hand, Multiple Input Multiple Output (MIMO) concept has been found in the literature to enhance either the transmission capacity or the link robustness (independent or dependent information streams are transmitted via parallel sub-channels simultaneously). In contrast, in MIMO radar there is no need to construct parallel subchannels which are fully dependent on a multipath environment. This is due to that all transmitted information is known at the receiver side. Thus, the channel matrix is used only for the purpose of sensing the environment, as an example to determine the number of the targets, their locations and velocities [9]-[12].

In order to enhance the MIMO radar, which is adopting the OFDM technique, a new work has been proposed in this paper based on hard decision-based PWM technique to tackle one of the main deficiencies found in OFDM; namely PAPR.

This deficiency appears due to the addition process with different frequencies and phases of numerous waves, which leads to the need of high dynamic ranges transmitters. The predicted PAPR values in OFDM signal can be formed as [7]:

$$\text{PAPR} = \frac{\max_{0 \leq t \leq T} (|\mathbb{Z}(t)|^2)}{\left(\frac{1}{T} \int_0^T (|\mathbb{Z}(t)|^2) dt \right)} \quad (1)$$

Here, $\mathbb{Z}(t)$ is the transmitted OFDM symbol and could be found as $\mathbb{Z}(t) = \sqrt{\frac{\left(\sum_{k=0}^{N-1} (X_k \cdot e^{(j2\pi f_k t)}) \right)^2}{N}}$,

which results from the modulation process of an N symbols data block; X_k with f_k , which are a set of orthogonal subcarriers for $k = 0, \dots, N-1$. The duration of OFDM symbol is denoted by T , which used to maintain the orthogonality for all values of t less than or equal to T . Moreover, to maintain the total transmitted power, the term $(1/\sqrt{N})$ has been imposed.

Such deficiency causes transmission amplification and other circuitry limitations. Therefore, to overcome this problem, average signal power must be kept low to allow the transmission process of the higher average power to be in a fixed level. Then an improvement of the reception process will be attained based on improving the signal to noise ratio.

There are several propositions and techniques that either tackle the PAPR effects or address the linearity and power efficiency issues, such as filtering and clipping techniques; coding based techniques; artificial intelligence based techniques; and signal representation techniques as the envelope elimination and restoration techniques and the phase shifted sequence ones. This is in order to optimize a solution at the expense of several challenges, such as the degradation of the Bit Error Rate (BER); the decrement of the spectral efficiency due to the side information (SI) transmission; and the computational complexity [13]-[16].

This paper addresses the proposition of a new technique based on using the pulse width modulation (PWM) to overcome the PAPR problem effect. Consequently, the overall performance will be enhanced for the MIMO radar, which adopting the OFDM technique. As a result of considering the use of PWM, a basis of controlling the power electronics [17], an optimum solution will be provided to optimize the vital parameters of the existing work such as the speed, and the area. Moreover, the proposed work has been compared with either conventional OFDM systems in the literature or our previous published work in order to show the performance improvements before applying it to a MIMO-OFDM radar system.

PWM signal is easily generated by comparing the reference signal with a carrier one. Mainly, the input signal is used to determine the width of the generated PWM signal. This is clearly shown in the following mathematical representation

$$\text{PWM}(t) = \text{sgn}(r(t) - c(t)) \quad (2)$$

where the generated PWM signal depends on the sign function of the subtraction process between the compared

reference signal; $r(t)$ and the carrier signal; $c(t)$.

As basic PWM signal generation, there are two methods that help in producing the variable pulses widths; direct digital generation and uniformly sampled PWM. They can be distinguished by the focusing on the controlling criteria. In this paper, the second technique will be chosen, where a triangle clock signal is used to generate the uniformly sampled PWM signal. This is due to that it does not need high frequency clock signal. Moreover, the triangle clock signal is chosen over the other two types, Sawtooth or the inverted Sawtooth, due to that it has low number of dominant higher harmonics. The achieved benefit here concluded in reducing the needed system bandwidth [17] [18].

The rest of this work is introduced as follows: Section 2 describes the model of MIMO-OFDM radar signals based on PWM along with the analytical formulation in addition to the computational complexity. Section 3 presents simulation results and hardware implementations; finally, the conclusion is represented in Section 4.

2. MIMO-OFDM Radar Signal Model-Based PWM

2.1. MIMO-OFDM Radar Systems Structure

In [1], OFDM technique has the advantage of combating the frequency selective fading drawback for a narrow-band system. This turns the researchers toward making use of such advantage to be imposed in MIMO radar systems. Therefore, the radar system performance will be improved by performing the target localization separately. This will be attained by making use of the combination of different orthogonal and narrowband sub-signals. As a result, the frequency selective fading deficiency is overcome by the frequency diversity utilization. Moreover, the complexity of MIMO-OFDM radar transmitters will be maintained at low level, since the used sub-band waveforms designed to have same characteristics as the narrowband MIMO radar ones.

The baseband MIMO-OFDM transmitted matrix is defined in (1) by $\mathbb{Z}(t)$, where an N sets of orthonormal signals have been sent simultaneously. The digital-to-analogue convertor (DAC) has been used to efficiently generate each OFDM symbol from $\mathbb{Z}(t)$ after the IFFT stage. For practical implementation using the IFFT, $\mathbb{Z}(t)$ should be oversampled [1] as shown below:

$$\left[\mathbb{Z}_{i,j}(1) \cdots \mathbb{Z}_{i,j}(L) \right] = \text{IFFT} \left(\left[\mathbf{X}_{i,j}(0) \cdots \mathbf{X}_{i,j}(k-1) \mathbf{0}_{1 \times (L-k)} \right] \right) \quad (3)$$

Here, the IFFT of $\mathbb{Z}(t)$ matrix of the size of $(i \times j)$ will have an L samples and defining its l -th sampling by $\mathbb{Z}_{i,j}(l)$, the element of i -th row and j -th column of the matrix $\mathbf{X}(k)$ is given by $\mathbf{X}_{i,j}(k)$, and $\mathbf{0}_{1 \times (L-k)}$ for zero padding when $L \geq k$. Accordingly, the produced i -th OFDM symbol will be processed and transmitted from the i -th antenna.

The next step after generating the OFDM baseband waveforms is the imposing of a guard interval process at the beginning of each OFDM symbol. This is attained by attaching a copy of the later OFDM symbol part at the beginning; namely a cyclic prefix process, which is introduced in order to maintain the orthogonality between the used sub-bands by MIMO operation and will be accomplished by making use of windowing techniques. Moreover, the MIMO-OFDM radar will make use of it in order to compensate for the shifts in time. This is clearly shown under the case of multiple targets at different ranges, where it guarantees the existence of the needed phase delay information inside the used window. This is true under a predefined separation range, which is based on the antenna array dimension and the used lengths for the transmitted symbol period and the cyclic prefix length. Therefore, the final transmitted baseband matrix is given as

$$\mathbb{Q}(t) = \mathbb{Z}(t - T_p) W \left(\frac{t - T_p}{T_s} \right) + \mathbb{Z}(t + T_s - T_p) W \left(\frac{t}{T_p} \right) \quad (4)$$

Here, T_p is the length of the cyclic prefix, T_s is the $\mathbb{Z}(t)$ elements period and equals $1/f_k$, W is a window function that has a value of unity when $t \in [0, 1]$.

Moreover, in order to maintain the orthogonality condition, the antenna elements displacement should satisfy certain threshold. In this work, to detect and estimate a target within 180° , the displacement; D should satisfy

$$D \leq \left(c \times (T_p - T_{\text{target}}) \right) \quad (5)$$

where, c is the speed of light and T_{target} is the target impulse response length.

The imposing process of the cyclic prefix is clearly described in **Figure 1**, where a simultaneous transmission of three OFDM symbols from three antennas is accomplished.

In this work, and after the imposing of the guard interval; *i.e.* the cyclic prefix, a new processing block has been inserted to analyze the PAPR performance. This is to free the channel from the inter symbol interference (ISI) drawback. Moreover, this choice will reduce the hardware area under the consideration of hardware implementation.

2.2. Proposition of PWM

Returning to the implementation of the baseband OFDM signal; *i.e.* $\mathbb{Z}(t)$; in (1), and the possibility of producing high PAPR values. The overall system performance will be enhanced if the PAPR effect is efficiently reduced. This work focusing on proposing PWM based work that will produce constant amplitude signals by making use of the consecutive samples slope. It's considered as a novel technique that is attaining the maximum power amplifier efficiency issues, which will permit the ability of using nonlinear devices easily. Complementary cumulative distribution function (CCDF) curves give the statistical characteristics of the PAPR distribution in OFDM systems [19] [20]. CCDF curves show the probability of exceeding the PAPR a certain threshold for different signal to noise ratio values, *i.e.* the maximum power amplifier efficiency will be attained at the minimum CCDF value.

The proposed PWM work starts with reshaping the signal, $\mathbb{Z}(t)$, in (1) as blocks, each with length equals to the IFFT points as follows

$$\mathbb{Z}(n, m) = \left(\sum_{k=0}^{N(m)-1} [\mathbf{X}(k, m)] \left[e^{j \frac{2\pi kn}{N(m)}} \right] \right) / \left(\sqrt{N(m)} \right) \quad (6)$$

Here, m stands for the block index as defined in (7), and $N(m)$ denotes the block length.

$$m = \begin{cases} 0, & n < N(0) \\ \rho, & \sum_{i=0}^{\rho-1} N(i) \leq n < \sum_{i=0}^{\rho} N(i), \rho \geq 1 \end{cases} \quad (7)$$

The reshaped result from (6) has been used to be processed in the production of a constant amplitude signal based on the PWM technique. This is clearly depicted and shown in **Figure 2**.

As shown in **Figure 2**, the conversion process is divided into the following stages:

First stage:

- In order to distinguish each symbol after the conversion process an extra zero sample has been added at the beginning of each OFDM symbol.
- The sampling rate has been increased in order to enhance the accuracy of the conversion process; $\mathbb{Z}(n, m)$ will have extra samples under the use of new sampling rate $\tilde{N}(m)$. This is clearly shown in (8) where $\mathbb{Z}(n, m)$ has been converted into a three vectors-matrix.

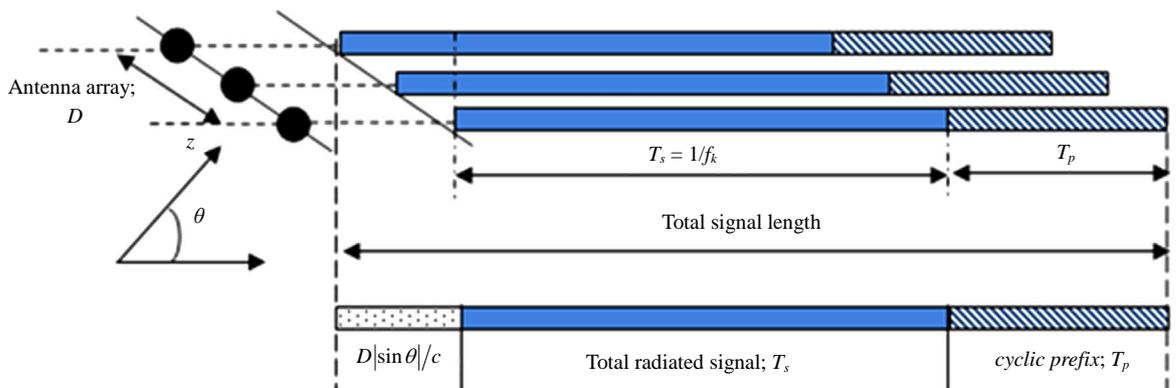


Figure 1. 3-MIMO-OFDM symbols transmission process.

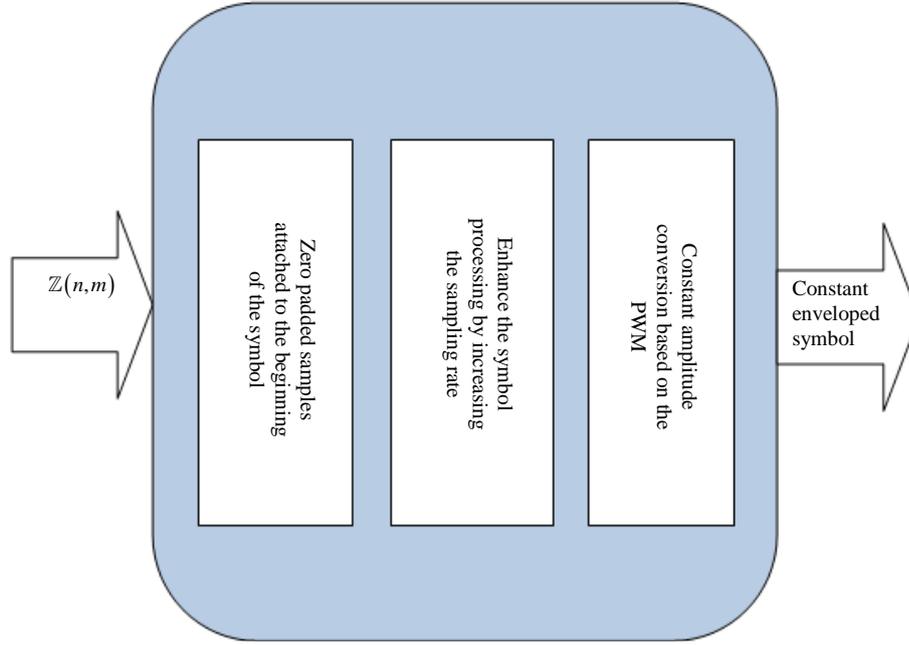


Figure 2. The conversion procedure.

$$\mathbb{Z}(3,i) = \begin{cases} \mathbb{Z}(n,m) \\ \mathbb{Z}(n,m) - \frac{1}{1 + \hat{N}(m)} \\ \mathbb{Z}(n,m) + \frac{1}{1 + \hat{N}(m)} \end{cases}, \quad (8)$$

where i stands for the sample value; $i \in [1, \hat{N}(m) + N(m)]$.

Second Stage:

In this stage, the oversampled version of the OFDM symbol will be processed in order to produce a constant envelope version. For simplicity, the slope between two consecutive samples has been chosen as a comparison criterion. This criterion is depicted as follows in (9).

$$\mathbb{Z}(i) = \begin{cases} \mathbb{Z}(i) = \mathbb{Z}(1,(i-1)), & \text{then } \mathbb{Z}(i) = 0 \\ \mathbb{Z}(i) < \mathbb{Z}(2,(i-1)), & \text{then } \mathbb{Z}(i) = -\text{slope} \\ \mathbb{Z}(i) > \mathbb{Z}(3,(i-1)), & \text{then } \mathbb{Z}(i) = +\text{slope} \end{cases} \quad (9)$$

These two stages are clearly described in Figure 3. It's divided into two main parts denoting the stages consequently.

As described earlier, Figure 3(a) depicts the process of the first stage and Figure 3(b) represents the procedure of the second stage. In Figure 3(b), the pre-process is divided into two main parts; the one that is responsible for simplifying the distinguishing process by adding a pre-known sample(s) at the beginning of the conventional symbol. In this work, the zero sample will do the expected results in either the transmission part or the reception one.

This is clearly found in Figure 4, where a zero sample has been attached at the beginning of the reshaped signal, i.e. $\mathbb{Z}(n,m)$. The second part deals with sampling rate of the modified $\mathbb{Z}(n,m)$; here extra samples have been imposed between the two consecutive samples by enhancing the sampling rate to be $\hat{N}(m)$. Figure 5, shows the enhancement sampling rate for the signal with 108 samples instead of 12 samples.

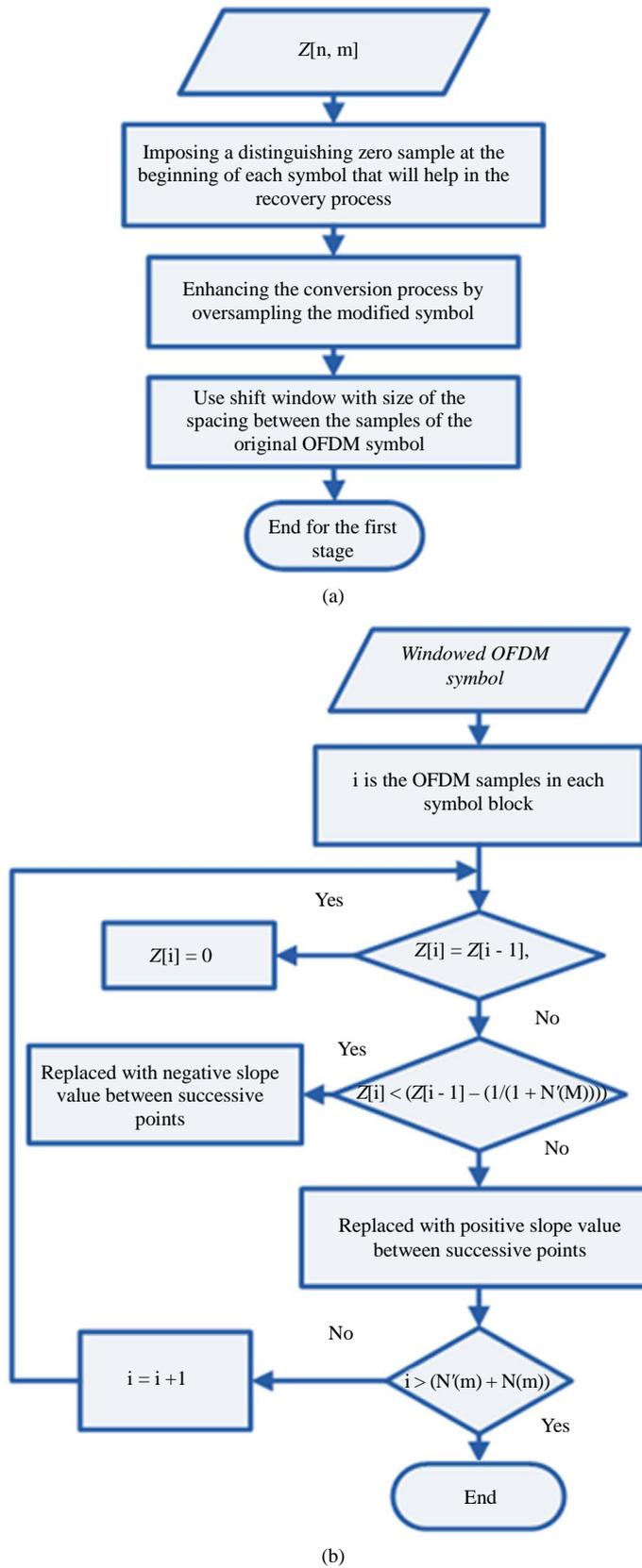


Figure 3. The proposed flowchart (a) First stage description; (b) Second stage description.

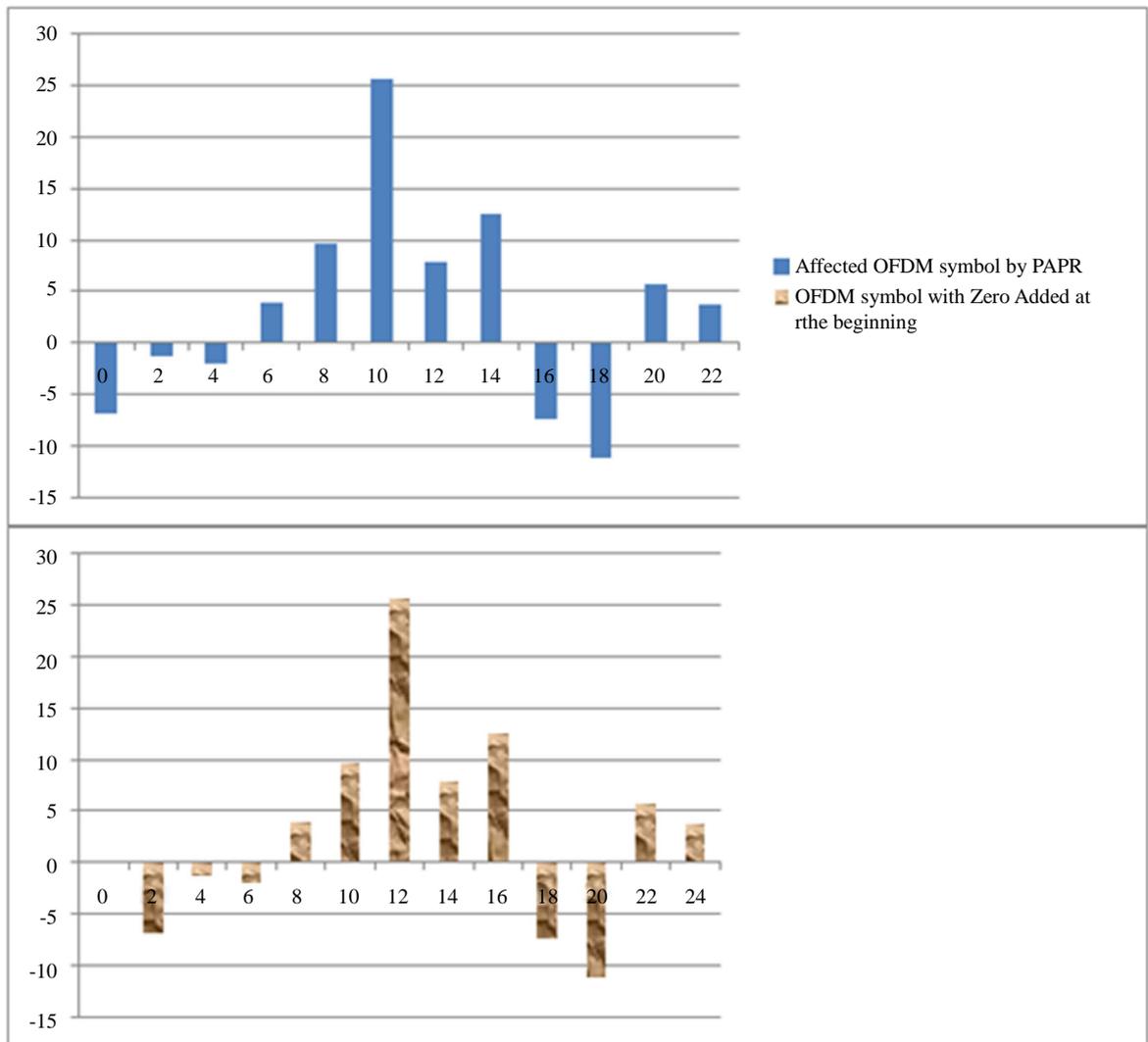


Figure 4. The process of adding the distinguishing zero sample.

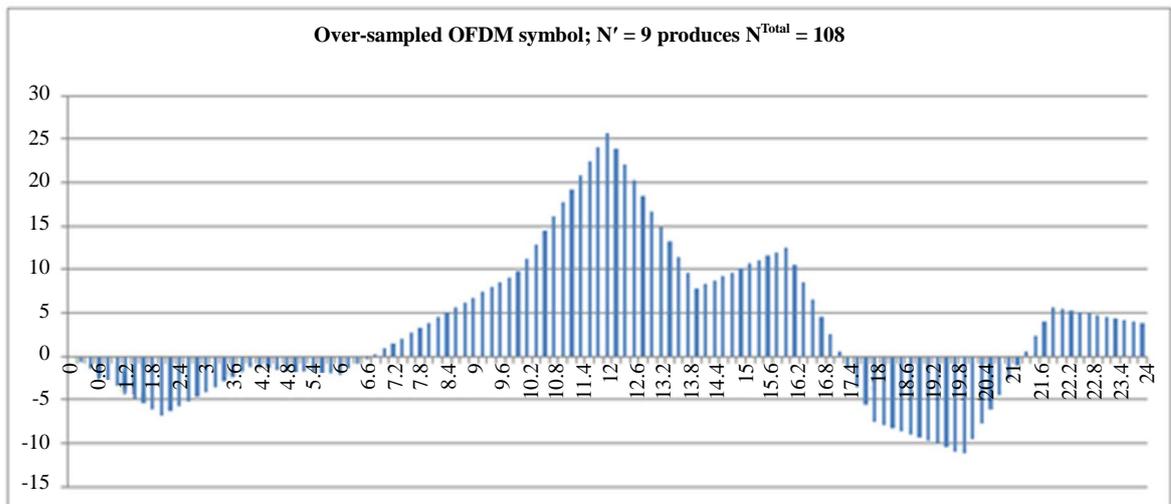


Figure 5. The over-sampled OFDM symbol.

Figure 6 depicts the output results that are achieved from the second stage in the proposed work and shown in Figure 3(b), which is drawn from the proposed comparison formula in (9). In this figure, the idea of the proposed work has been showed up; *i.e.* the OFDM symbol has been converted to a constant enveloped symbol based on the slope between the adjacent samples.

The variation has been reduced and the peaks values have been diminished, consequently the PAPR values will be reduced. In addition, to simplify the proposed work, the hermitian structure of the OFDM systems could be exploited [21]. At this stage, the OFDM symbol is ready to be transferred to the next block; MIMO block to process the rectified symbol based on the Vertical-Bell Laboratories Layered Space-Time (V-BLAST) criterion. V-BLAST technology is used to attain the system capacity/throughput enhancement, which is expressed in terms of bits/symbol.

The transmitter stages are shown in Figure 7. The transmitted signal through I antennas will be guaranteed to be with the minimum PAPR value, since the resultant unaffected MIMO-OFDM signal will be based on the constant enveloped transformed signal using PWM. As a result, the novelty of this work rises from the way of dealing with the OFDM symbol for such application. This is in addition to the way of how to reduce the CCDF curves values that are guaranteed to remains at their minimum levels.

In the receiver side, the signal modelling will be determined based on the sent and received signals between/ among the transmitter and the object(s), which will help in determining the objects specifications. The issues of determining the objects directions and locations are considered out of scope of this work and will be discussed in another work. Thus, it is focusing on how to overcome the rise problem due to the use of the FFT and its inverse in modelling the OFDM system.

After the transmission through a channel from different transmitting antennas, in the receiver side, the main task of the receiver is to recover the original OFDM signal from the modified one. Accordingly, the used recovering

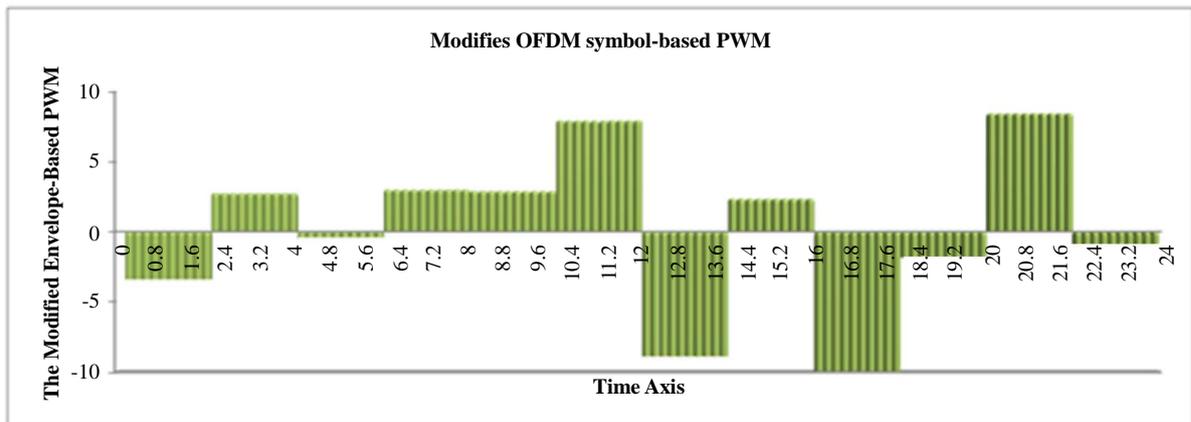


Figure 6. Modified OFDM symbol based PWM.

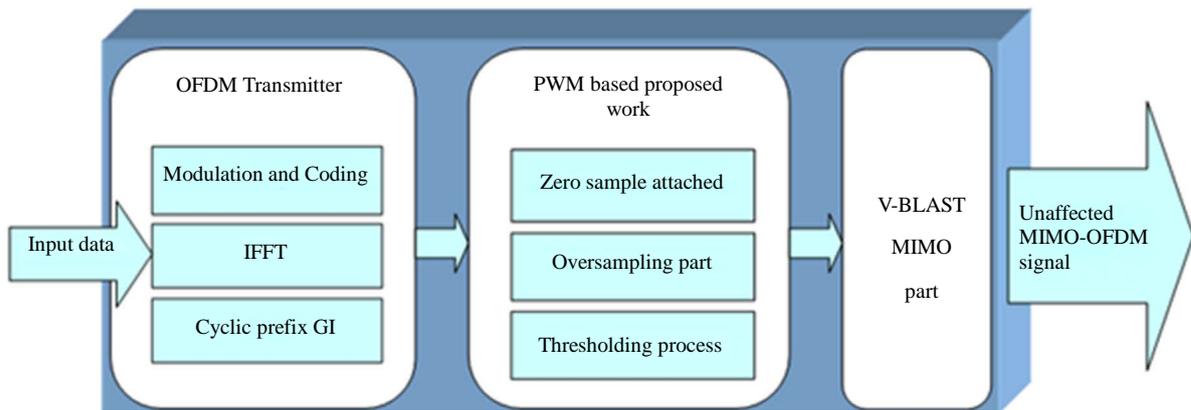


Figure 7. The proposed MIMO-OFDM radar transmitter.

procedure will be divided into two main stages; firstly, proposing an algorithm to recover the OFDM symbol from the constant enveloped received symbols, and secondly, a signal processing stage based on removing the extra imposed samples. This procedure is clearly shown and described in **Figure 8**. It contains two parts depicted the proposed processing stages; **Figure 8(a)** and **Figure 8(b)** according to stage 1 and stage 2 consequently.

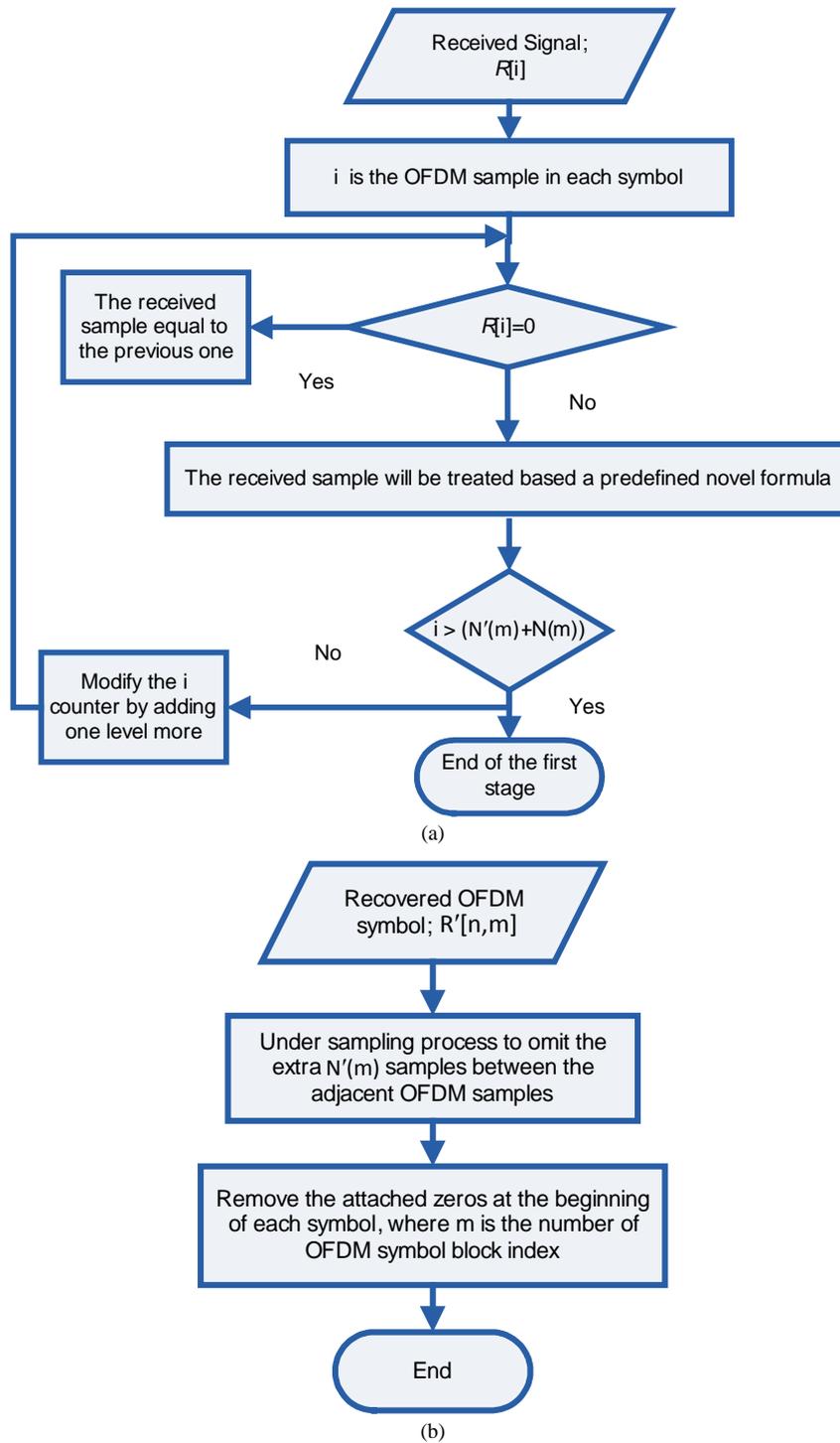


Figure 8. Reception stage flowchart (a) First stage procedure; (b) Second stage procedure.

The reproduction process of regenerating OFDM signal from the received is clearly depicted in **Figure 8**. It is divided into two main parts; the reconversion stage and the de-processing stage; shown in **Figure 8(a)** and **Figure 8(b)** respectively.

The given procedure in **Figure 8(a)** starts with checking the received samples to clarify whether it is the beginning of new symbol or not as shown in (10). Moreover, a novel formula has been proposed to reproduce the generation of the original OFDM symbols based on three different variables; the received sample, the difference in time between a consecutive samples and the previous sample.

$$R[i] = \begin{cases} 0 \Rightarrow \left\{ \begin{array}{l} \text{if } R[i-1] \left\{ \begin{array}{l} \text{existed} \\ \text{does not exist} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} R[i] = R[i-1] \\ \text{new symbol starts} \end{array} \right. \\ \\ \text{O.W.} \left\{ \begin{array}{l} \text{the sample will be treated based on a novel formula,} \\ \text{which is depending on the received sample and the time} \\ \text{interval and the previous received sample} \end{array} \right. \end{array} \right. \quad (10)$$

Furthermore, in **Figure 8(b)** the conversion process has been completed by fulfilling the removal of the extra added samples. The novelty in this work has been shown in how to deal with the OFDM symbol; imposing hard decision criterion based on the PWM instead of sending the original OFDM signals.

The next section describes the results from the proposed work simulation against the conventional techniques; they are based on both of CCDF and SER curves. These two criteria are used to validate the OFDM systems performance, where the lower the values the higher the system performance.

3. Tested System Performance and Simulated Results

The proposed MIMO-OFDM radar system that has been described in **Figure 7** is used to localise a composite target based on a four-element array. The composite target has been placed far away from the array, which contains three dielectric elements. They are differed in physical dimensions that are given in terms of the carrier frequency but have same dielectric permittivity. Moreover, the used array is assumed to contain isotropic elements with equal spacing. The scope of this work covers the system performance based on tackling the PAPR problem in the MIMO-OFDM radar. This leaves a room to enhance the performance from different point of views for future work.

At this stage, the performed MATLAB simulation has the following specifications:

- Carrier frequency of 3 GHz,
- Extra 9 samples have been added between the consecutive samples; $N'(m) = 9$,
- IFFT length of 1024 point,
- $T_p = 0.25 \times T_s$,
- Carrier spacing = (1/32) GHz,
- 64-QAM modulation technique,
- Vertical-Bell Laboratories Layered Space-Time (V-BLAST) MIMO system will be used for the four-element array transmission. This is in order to boost the system performance in terms of bits/symbol.

To imitate a real scenario, the shown proposed work in **Figure 7** has been imposed twice; one for the real OFDM symbol part and the other one for the imaginary part. Moreover and in order to test the system's performance, the results have been divided into two parts; the complementary cumulative distribution function (CCDF) part and the sample error rate (SER) part. **Figure 9** will depict the value of proposed work SER and how promising the achieved values are, while **Figure 10** will clarify the enhancement in combating the effect of the PAPR from the CCDF point of view, which is considered as a performance metric independent of the transmitter amplifier. It is defined by how often the PAPR is higher than a given threshold; PAPR_o . It is expressed by:

$$\text{CCDF}(\text{PAPR}_o) = \text{Probability} \{ \text{PAPR} > \text{PAPR}_o \} \quad (11)$$

As depicted in **Figure 9**, the proposed PWM enhances the recovery process of the original OFDM samples. This work is divided into two experiments; the first one based on using the previous sample in order to predict the received sample, while the other one is based on using the average of all samples within the window. The one that is based on the average gives a better SER under the optimization issue between the SER and the delay

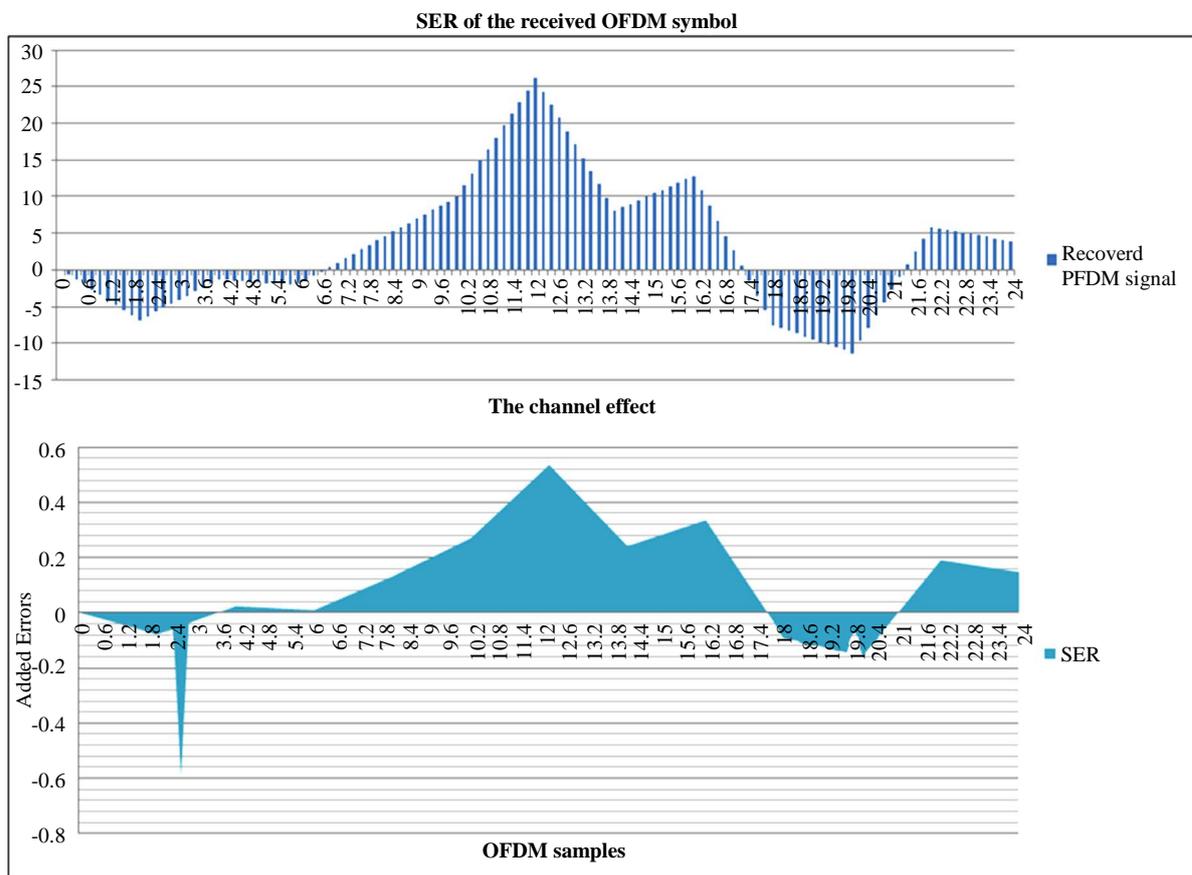


Figure 9. The SER and the recovered OFDM samples.

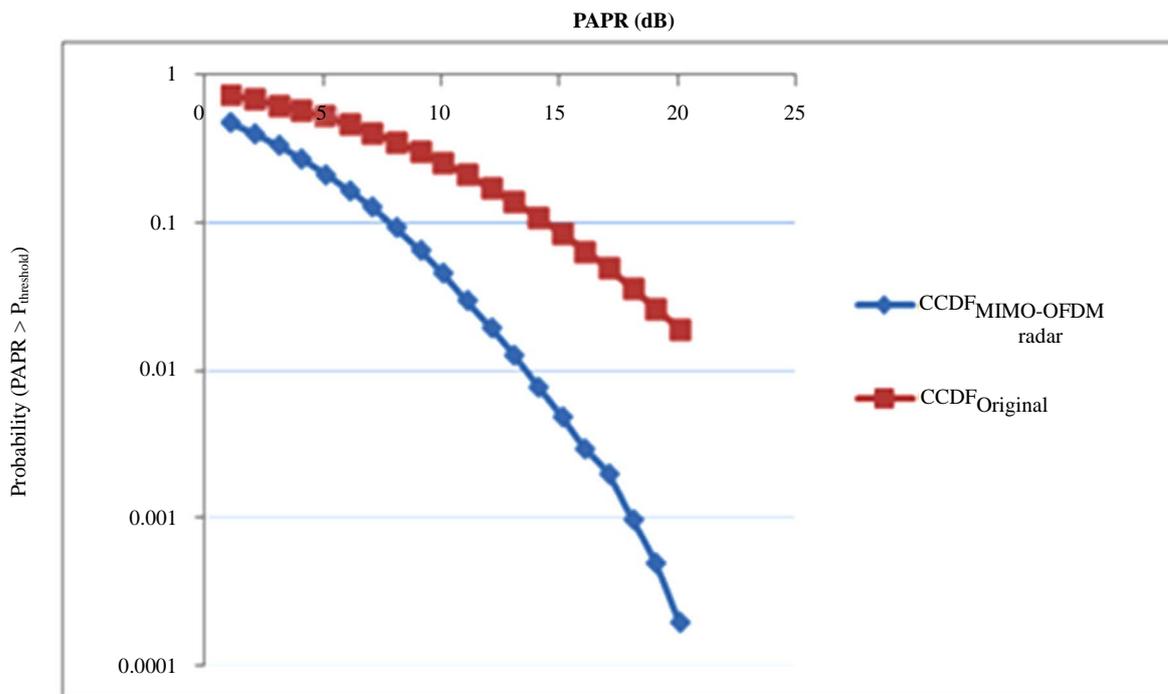


Figure 10. The CCDF results based on 64QAM modulation technique.

Table 1. The Simulation results of the proposed technique based on PWM to the literature work.

| Modulation technique | CCDF (2%) | | Additional reduction (%) | | |
|----------------------|--------------------------|--------------------------------|--------------------------|------|-----|
| | PAPR without coding (dB) | MIMO-OFDM radar based PWM (dB) | Clipping | SLM | PTS |
| 64 QAM | 20 | 11.5 | 72.1 | 44.5 | 11 |

time. This modification enhances the SER from 9.3×10^{-4} to 8.7×10^{-4} .

Furthermore, the second part of the systems performance checking is shown in **Figure 10** based on the CCDF curves. From the depicted results in **Figure 10**, the proposed work gives a noticeable improvement in the CCDF curves. Using the PWM as a processing technique to enhance the OFDM signal's envelope reduces the CCDF (20 dB) from 19×10^{-3} to 2×10^{-4} . Furthermore, a comparison has been made between our proposed work and the work that found in the literature. This comparison is shown in **Table 1**.

The proposed system performance improvement has been clearly shown in **Table 1**. It shows that our work has 2% of the PAPR that exceeds 11.5 dB, where for the same probability percentage the conventional OFDM systems has a PAPR over 20 dB. Moreover and comparing to the literature, the proposed work has an extra reduction percentage between 11% and 72%.

MIMO-OFDM radar work is different from the conventional ones in the literature, where this work has built the comparison stage making use of the slope between the two consecutive samples. Additionally, this work enhances the use of the PWM techniques, where the conventional PWM links the comparison performance to the inserted number of extra samples. In the MIMO-OFDM radar, the performance has been improved without overloading the systems with extra samples since the comparison stage has been linked to the slope between the samples.

4. Conclusions

This work takes high PAPR effect into consideration when proposing the OFDM technique to the conventional MIMO radar systems. High PAPR values could reduce the system performance especially when using nonlinear devices. A new work has been proposed to overcome this deficiency making use of the conventional PWM with some modifications.

A MATLAB simulation has been conducted to validate the analytical model of the proposed work. It consists of two parts: one to check the sample error rate (SER) after the recovery process, while the other one to describe the probability of the PAPR to exceed a certain threshold. Moreover, a comparison with the literature has been made in order to confirm the expected performance modification.

Under same environmental conditions and system specifications, a SER of 8.7×10^{-4} has been achieved compared with the transmission of conventional OFDM signals. This is in addition to enhancing the probability of the PAPR that exceeds 20 dB from 2.1×10^{-2} to 1.7×10^{-4} . The work validity has been checked based on a comparison with the ones in the literature, such as PTS, SLM or Clipping techniques, the proposed work gives an additional PAPR reduction percentage between 11% and 72% over the achieved 11.5 dB value. As a consequence, the transmission throughput will improve.

References

- [1] Nee, R. and Prasad, R. (2000) OFDM for Wireless Multimedia Communications. Artech House, Norwood.
- [2] Umali, E., Toyama, Y. and Yamao, Y. (2008) Power Spectral Analysis of the Envelope Pulse-Width Modulation (EPWM) Transmitter for High Efficiency Amplification of OFDM Signals. *IEEE Vehicular Technology Conference (VTC)*, Singapore, 1261-1265.
- [3] 3GPP, Tech. Specif. Group Services and System Aspects Service Requirements for Evolution of the 3GPP System (Rel. 8), 3GPP TS 22.278.
- [4] Dahlman, E., *et al.* (2008) 3G Evolution: HSPA and LTE for Mobile Broadband. 2nd Edition, Academic Press.
- [5] Abeta, S. (2010) Toward LTE Commercial Launch and Future Plan for LTE Enhancements (LTE-Advanced). *IEEE International Conference on Communication Systems (ICCS) Proceedings*, Singapore, 146-150.
- [6] Andrews, J., Ghosh, A. and Muhamed, R. (2007) Fundamentals of WiMAX: Understanding Broadband Wireless Networking. Prentice Hall.

-
- [7] Jiang, T. and Yu, Y. (2008) An Overview: Peak-to-Average Power Ratio Reduction Techniques for OFDM Signals. *IEEE Transaction on Broadcasting*, **54**, 257-268. <http://dx.doi.org/10.1109/TBC.2008.915770>
- [8] Saeed, A., Elbably, M. and Abdelfadeel, G. (2009) Efficient FPGA Implementation of FFT/IFFT Processor. *International Journal of Circuits, Systems and Signal Processing*, **3**, 103-110.
- [9] Foschini, G. and Gans, M. (1998) On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas. *Wireless Personal Communications*, **6**, 311-335. <http://dx.doi.org/10.1023/A:1008889222784>
- [10] Telatar, E. (1995) Capacity of Multi-Antenna Gaussian Channels'. AT & T Bell Laboratories.
- [11] Zelst, V. and Schenk, T. (2004) Implementation of a MIMO OFDM-Based Wireless LAN System. *IEEE Transactions on Signal Processing*, **52**, 483-494. <http://dx.doi.org/10.1109/TSP.2003.820989>
- [12] Bekkemani, I. and Tabrkian, J. (2006) Target Detection and Localization Using MIMO Radars and Sonars. *IEEE Transactions on Signal Processing*, **54**, 3873-3883. <http://dx.doi.org/10.1109/TSP.2006.879267>
- [13] Wang, Y. and Luo, Z. (2011) Optimized Iterative Clipping and Filtering for PAPR Reduction of OFDM Signals. *IEEE Transactions on Communications*, **59**, 33-37. <http://dx.doi.org/10.1109/TCOMM.2010.102910.090040>
- [14] Chen, J.C. (2010) Partial Transmit Sequence for PAPR Reduction of OFDM Signals with Stochastic Optimization Techniques. *IEEE Transactions on Consumer Electronics*, **56**, 1229-1234. <http://dx.doi.org/10.1109/TCE.2010.5606251>
- [15] Sohn, I. (2014) A Low Complexity PAPR Reduction Scheme for OFDM Systems via Neural Networks. *IEEE Communications Letters*, **18**, 225-228. <http://dx.doi.org/10.1109/LCOMM.2013.123113.131888>
- [16] Wang, F.P., Kimball, D., Popp, J., Yang, A., Lie, D., Asbeck, P. and Larson, L. (2005) Wideband Envelope Elimination and Restoration Power Amplifier with High Efficiency Wideband Envelope Amplifier for WLAN 802.11g Applications. *IEEE Microwave Symposium Digest*, Long Beach, 12-17 June 2005, 645-648.
- [17] Vasca, F. and Lannelli, L. (2012) Dynamics and Control of Switched Electronic Systems: Advanced Perspectives for Modelling, Simulation and Control of Power Converters. Springer Publisher, Berlin.
- [18] Koyuncu, M., van den Bos, C. and Serdijn, W. (2000) A PWM Modulator for Wireless Infrared Communication. *Proceedings of the ProRISC/IEEE Workshop on Semiconductors, Circuits, Systems and Signal Processing*, Veldhoven, 30 November-1 December 2000, 351-353.
- [19] Ochiai, H. and Imai, H. (2001) On the Distribution of the Peak-to-Average Power Ratio in OFDM Signals. *IEEE Transactions on Communications*, **49**, 282-289. <http://dx.doi.org/10.1109/26.905885>
- [20] Wei, S., Goeckel, D. and Kelly, P. (2010) Convergence of the Complex Envelope of Bandlimited OFDM Signals. *IEEE Transactions on Information Theory*, **56**, 4893-4904. <http://dx.doi.org/10.1109/TIT.2010.2059550>
- [21] Lin, H. and Siohan, P. (2008) OFDM/OQAM with Hermitian Symmetry: Design and Performance for Baseband Communication. *IEEE International Conference on Communications (ICC' 08)*, Beijing, 19-23 May 2008, 652-656.

Dynamic Capacity Allocation in OTN Networks

Maria Catarina Taful¹, João Pires^{1,2}

¹Department of Electrical and Computer Engineering, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal

²Institute of Telecommunications, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal
Email: mariacatarinataful@gmail.com, jpires@lx.it.pt

Received 23 January 2015; accepted 12 February 2015; published 15 February 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

A dynamic Optical Transport Network (OTN) has the advantage of being able to adjust the connection capacity on demand in order to respond to variations on traffic patterns or to network failures. This feature has the potential to reduce operational costs and at the same time to optimize networks resources. Virtual Concatenation (VCAT) and Link Capacity Adjustment Scheme (LCAS) are two techniques that when properly combined can be used to provide improved dynamism in OTN networks. These techniques have been previously standardized in the context of Next Generation SDH/SONET networks. VCAT is used to tailor the capacity of network connections according to service requirements, while LCAS can adjust dynamically that capacity in a hitless manner. This paper presents an overview of the application of VCAT/LCAS techniques in the context of OTN. It explains in detail how these techniques can be employed to resize the connection capacity and analyses its use in network protection solutions. Furthermore, a detailed analysis of the time delays associated with different operations is provided and its application to some reference networks is undertaken. The obtained results provide an idea about the time delays of the capacity adjustment processes and define potential scenarios for implementing VCAT/LCAS techniques.

Keywords

OTN, VCAT, LCAS, Time Delay, Network Protection

1. Introduction

In order to support the constant growing of network traffic and the increasing heterogeneity of services/applications the transport infrastructure of telecommunication networks is facing a series of new challenges. The traffic growth imposes the usage of very-high bit rates (*i.e.* 40 and 100 Gb/s) and the rising variety of flows requests

for more flexible, reliable and dynamic network designs. These networks must be capable, for example, to provide fast re-provisioning of services to accommodate traffic fluctuations and at the same time to respond as quickly as possible to network failures. The Optical Transport Network (OTN) technology is expected to be the right technology to handle these challenges [1] [2]. It has been standardized by ITU-T in Recommendation G.709 [3], operates at layer 1 of the Open Systems Interconnection (OSI) communications model and it is itself subdivided into two layers: an electrical layer also called digital wrapper and an optical layer also called Dense Wavelength Division Multiplexing (DWDM) layer. The electrical layer is responsible for mapping client signals into entities called Optical Data Units (ODUs), as well as for multiplexing, switching and managing these entities, whereas the optical layer is responsible for generating, multiplexing, switching and managing optical channels. The ODU_ks ($k = 0, 1, 2, 3, 4$) are transport containers used to carry client signals between an end-to-end path and can be either of fixed or variable size. The containers of fixed size are standardized to support certain client signals. For example, ODU₄ is intended to transport a 100 GbE signal. To obtain a container of variable size, there are two techniques available: Flexible Rate ODU (ODUflex) and Virtual Concatenation (VCAT) [4] [5].

In ODUflex, a certain number of Tributary Slots, each one with a granularity of approximately 1.25 Gb/s, are combined and the resulting structure is mapped into a fixed higher order ODU_k to be transported as a single entity. On the other hand, VCAT is an inverse multiplexing technique, by which each payload container of a given traffic flow is segmented into smaller containers, which are logically combined to form a Virtual Concatenation Group (VCG) and transported independently of each other over the same route (single-path routing) or over different routes (multipath-routing) [6].

In order to adjust in a flexible mode, the capacity allocated to connections, the network must be capable of dynamically changing the size of the containers in a hitless manner, *i.e.* without affecting the service. The resizing of ODUflex containers can be accomplished using the protocol Hitless Adjustment of ODUflex (HAO), while the members of a VCG can be added or removed through the Link Capacity Adjustment Scheme (LCAS). Both techniques have their own advantages and drawbacks. ODUflex is easier to implement and manage than VCAT and, as each signal is transported as a single entity, it does not require differential delay compensation, as it is required with the second technique. However, resizing operations are more complex for ODUflex paths than for VCAT ones, since they require the participation of all nodes in the path, contrary to VCAT, where only the ingress and egress nodes take action in the operation. Furthermore, when multipath routing is provided, the scheme based on Virtual Concatenation permits to implement traffic engineering techniques, such as load balancing, guaranteeing the use of network resources more efficiently [7]. In addition, LCAS can be employed for resilience purposes [8]-[11]: it can automatically remove disrupted VCG members in the presence of link failures, assuring that an unprotected ODU connection still continues operating despite working at a lower capacity; it can also be employed to activate backup VCG members used in protected connections whenever necessary. The first scheme is particular useful in data communications with unprotected connections where it is preferable to have a connection working at lower bit rate with “degraded service”, rather than no connection at all. The second scheme rely on the existence of backup VCG members, which are set up in advance using paths which are link disjoint from the working ones, to protect the working members.

In traffic engineering applications, the reconfiguration time of a VCG is not a crucial issue as far as the operation does not take place in real time. However, in protection applications, it is important to be able to calculate the time required for adding or removing VCG members from a connection in order to compute the fault recovery time, *i.e.* the time elapsed between the instant a failure is detected and the instant the traffic is recovered.

This paper focus on the problem of VCG reconfiguration in OTN networks by using LCAS and details how the reconfiguration time can be calculated considering some typical reference networks. Although the impact of LCAS on the dynamic bandwidth adjustment in the context of Next Generation (NG)-SDH/SONET networks has been previously analyzed [12], no similar analyses have been published on OTN networks to the best of our knowledge. Furthermore, we address the problem of evaluating the fault recovery time in the cases where the VCAT/LCAS is also applied for resilience purposes.

The rest of this paper is organized as follows. Section 2 reviews the operating principles of VCAT and LCAS technologies. Section 3 explains how the time required by LCAS to add or remove VCG members, as well as the fault recovery time, can be calculated. Section 4 adds some illustrative examples considering well-known reference networks and Section 5 concludes the paper.

2. VCAT and LCAS Overview

The OTN is designed to accommodate different client signals both at wavelength and sub-wavelength granularity [13]. The sub-wavelength operation is based on an electrical layered structure comprising the Optical Channel Payload Unit (OPU), Optical Channel Data Unit (ODU) and Optical Transport Data Unit (OTU). The OTU layer is the electrical content of the Optical Channel (OCh), which itself is the basic unit to be used when wavelength granularity is required. The VCAT in the OTN is realized by logically aggregating X OPU k ($k = 1, 2, 3$) signals. Note that Virtual Concatenation for OPU k with $k = 0, 2e, 4$, flex is not supported by the standard. The aggregated signal corresponds to the VCG and is denoted as OPU k -X v , where X is in the range from 1 to 256 and the lowercase v denotes Virtual Concatenation. The structure of the OPU k -X v frame is depicted in **Figure 1** using a bi-dimensional representation.

It consists of a matrix of octets with 4 rows and $X \times 3810$ columns, where columns $14X+1$ to $16X$ correspond to the OPU k overhead area and columns $16X+1$ to $3824X$ to the payload area (OPU k -X v column numbers are derived from the OPU k columns in the ODU k frame). The columns $14X+1$ to $15X$ include the Virtual Concatenation Overhead (VCOH) formed by the three octets VCOH1/2/3, which are used to carry the control information responsible for the VCAT process. It can also be referred that columns 1 to $14X$, which are omitted in **Figure 1**, correspond to the ODU k and OTU k overhead and, as a consequence, are only included in the OTN frame structure. The capacities of the different VCGs in an OTN network are shown in **Table 1** [6].

The presented results reveal that VCAT applied in OTN networks can achieve quite impressive capacities, much far beyond the ones that ODUflex can offer, since those are limited by the ODUs capabilities. Note, for example, that it will be possible to transport in the future a flow of about 10 Tb/s using an OPU3-X v , while without VCAT the use of an OPU3 only allows the transport of about 40 Gb/s.

The implementation of VCAT requires the usage of control signals which reside mainly on the OPU overhead. Two control signals are defined: the Multi-Frame Indicator (MFI) and the Sequence Number (SQ). The MFI is used to numerate each successive payload container frames of the traffic flow, in such a way that all OPU k s of the same VCG have the same MFI. On the other hand, each OPU k member of a VCG has its own and unique SQ, which is in the range from 0 to $(X-1)$. The sequence numbers are used by the destination node to reconstruct the original payload containers sequence having, in a certain way, a similar role to the sequence numbers used in the Real Time Protocol [14].

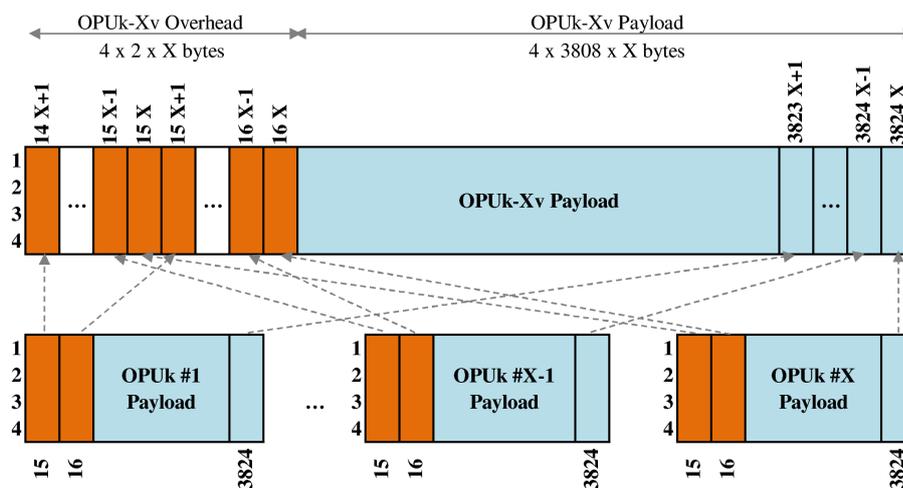


Figure 1. OPU k -X v frame structure (adapted from [3]).

Table 1. Capacities for different VCGs with VCAT.

| VCG type | X range | Capacity (Gb/s) |
|------------|----------|-----------------------|
| OPU1-X v | 1 to 256 | ~2.488 to ~637.010 |
| OPU2-X v | 1 to 256 | ~9.995 to ~2558.710 |
| OPU3-X v | 1 to 256 | ~40.151 to ~10278.533 |

As the VCAT employs a two-stage multi-frame, there is one MFI per stage. The MFI of the first stage uses the Multiframe Alignment Signal (MFAS) of the OTN frame alignment overhead area as an 8-bit indicator, and cycles from 0 to 255. As the MFAS is incremented by 1 every OPUk frame, the first stage multiframe (MFAS multiframe) has 256 OPUk frames.

The MFI of the second stage includes the MFI1 and MFI2 bytes to form a 16-bit indicator, which cycles from 0 to 65,535 since it is incremented at the start of each MFAS multiframe (MFAS = 0), thus it can take 65,536 different values. The bytes MFI1 and MFI2 are located in the first and second octet, respectively, of the Virtual Concatenation Overhead (VCOH1), while the bytes SQ are placed in the fifth octet, as shown in **Figure 2**.

The LCAS permits to change dynamically the size of a VCG by adding or removing members in a hitless manner with the operation being controlled by a network management plane, or by a control plane like GMPLS [15], or even by a network operating system using the OpenFlow protocol [16]. LCAS is implemented using a number of control signals, which also reside in the VCOH1 octet of the OPU overhead, with exception of the Member Status (MST) field which resides in the VCOH2 octet. From the source node to destination node, besides the MFI and SQ, LCAS also uses the Control (CTRL) word and the Group Identification (GID) bit. In the opposite direction, *i.e.* from the destination node to the source node, LCAS uses the MST field and the Re-Sequence Acknowledge (RS-Ack) bit. The CTRL field has the following states:

- FIXED: the number of members of the concatenated group cannot be changed (VCAT without LCAS);
- ADD: this member is going to be added to the concatenation group;
- NORM: this member is active and is used to transport data;
- EOS (End-Of-Sequence): this member is the last of the concatenation group;
- IDLE: this member is not part of the concatenation group or is in the process of being removed;
- DNU (Do-Not-Use): this member has a failed path to the destination node and must not be used. The backup members used to protect working members, while not active must be in this state.

The MST field is used to report the status of all the VCG members from the destination node back to the source node, using for that purpose a multi-frame defined by the last five bits of the MFAS signal (MST multi-frame), which are used to form a 5-bit indicator, which cycles from 0 to 31. The status of each member has two states:

- OK: This member is part of the concatenation group and has been correctly received at the destination;
- FAIL: This member is not part of the concatenation group, or has been received with failures.

In addition, GID identifies the VCG with all members of the same group having the same GID value, while RS-Ack is used by the destination to inform the source that it is aware of a change in SQ sequence of the VCG members.

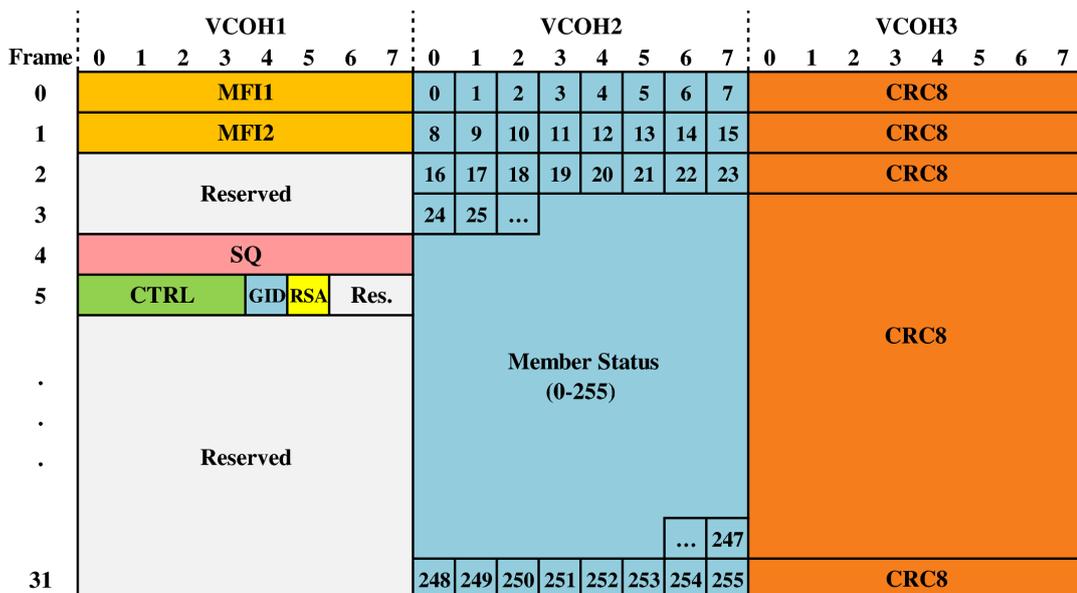


Figure 2. OPUk-Xv virtual concatenation overhead (adapted from [3]).

3. Time Delay of LCAS Operations

3.1. Components of the Time Delay

The time delay introduced by LCAS operations includes the multiframes propagation delay, the node processing delay and the LCAS message processing time. The first two terms are determined by the network physical topology. Assuming that a given path between a source and a destination node has a length l and passes through n intermediate nodes, the contribution of these terms is described by:

$$t_d = l \times \tau_f + n \times \tau_n, \quad (1)$$

where τ_f is the propagation delay per km and τ_n is the latency per node. For a typical optical fiber $\tau_f = 5 \mu\text{s}/\text{km}$ and the maximum latency per node is $\tau_n = 25 \mu\text{s}$ [9]. The LCAS message processing time depends on the operations being performed. In the following lines two cases will be analyzed: 1) The addition of a new member to a VCG, leading to an increase of the connection capacity; 2) The removal of an existent member from a VCG, leading to a decrease of the connection capacity. In both cases the processing time of LCAS depends on the time required to generate an MFAS multiframe (t_{MFAS}), the time to generate an MST multiframe (t_{MST}) and the number of multiframes exchanged to complete the adjustment. The presented results include, besides the LCAS message processing time, its propagation delay. The node processing delays are omitted for simplicity since they are negligible regarding the overall operation delays.

In OTN networks the duration of MFAS and MST multiframes is given by the duration of a single OPUk frame (t_{OPUK}) multiplied by the number of frames corresponding to the structure, which is 256 and 32, respectively. The duration of these multiframes is shown in **Table 2**. The MST multiframe requires 32 frames to transmit the state of all the members of the VCG OPUk-256v, because this information is transmitted in the byte VCOH2 of the OPUk overhead and, as a result, a single frame can only transmit the state of 8 VCG members.

3.2. Connection Capacity Increase

The process of increasing the capacity of a connection by adding new members to a given VCG (OPUk-Xv in OTN) is assumed to be initiated by the Network Management System (NMS). As shown in **Figure 3**, in an initial state the new member to be added has its CTRL command set to IDLE, since it is not yet a member of the VCG, its SQ number set to the maximum value supported and its MST value set to FAIL. In order to increase the connection capacity the NMS sends a request to the source node for the addition of a new member to the VCG. As a result the following operations take place:

1) The source node generates and transmits an MFAS multiframe, where the CTRL word of the member to be added to the VCG is changed from IDLE to ADD, as an indication to the destination node that the correspondent member is going to be added, together with the new assigned SQ (higher than the previous maximum used SQ). This corresponds to the time to generate and transmit an MFAS multiframe plus the time of its propagation to the destination node ($2t_{\text{MFAS}} + t_d$).

2) After the destination node having received the CTRL = ADD it sends back an MST multiframe with the status of the new member changed to OK. Thus the delay of this action is the time needed to transmit an MST multiframe plus the time of its propagation to the source node ($t_{\text{MST}} + t_d$).

3) Once the source receives this acknowledgment it generates and transmits a new MFAS multiframe with the new member added. The new member has now its CTRL set to EOS and its SQ assigned, so the decoder can know the correct order of the members when they arrive to destination node. This action takes the time to generate and transmit an MFAS multiframe plus the time of its propagation to the destination node ($2t_{\text{MFAS}} + t_d$). In this analysis it is considered that if a new CTRL code is to be inserted while a given MFAS multiframe is being transmitted, the new control code can only be inserted after finishing the multiframe transmission, whence it is used ($2t_{\text{MFAS}}$) in the calculations [12].

Table 2. MFAS and MST multiframe duration.

| | OPU1 | OPU2 | OPU3 |
|-------------------|----------------------|-----------------------|----------------------|
| t_{OPUK} | 48.971 μs | 12.191 μs | 3.035 μs |
| t_{MFAS} | 12.537 ms | 3.121 ms | 0.777 ms |
| t_{MST} | 1.567 ms | 390.112 μs | 97.120 μs |

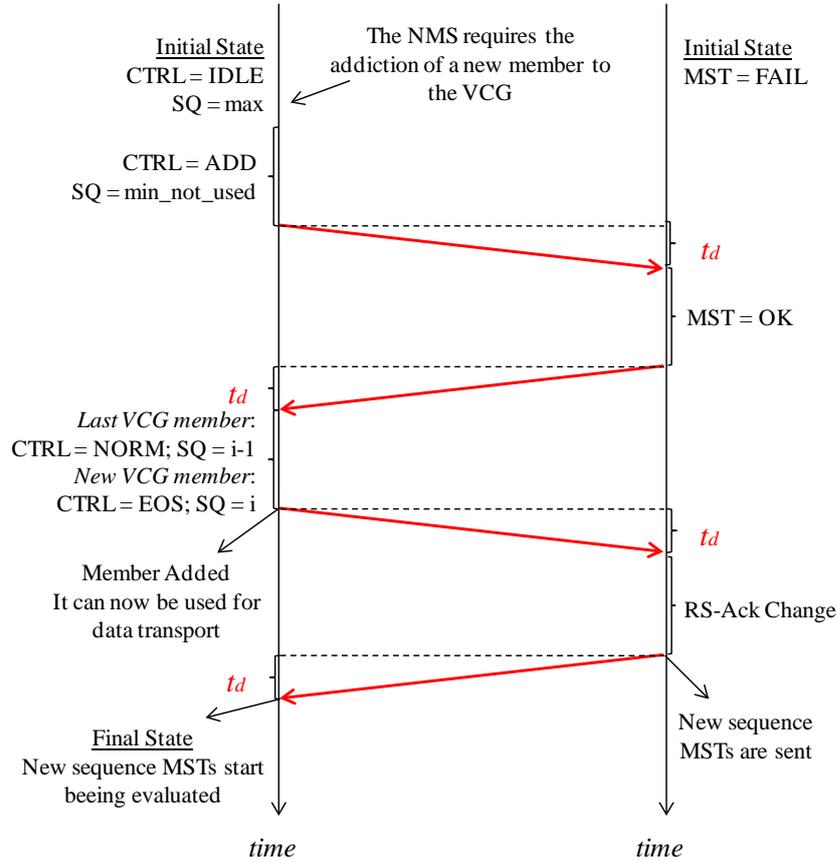


Figure 3. Time diagram for LCAS capacity increase process.

4) When the destination node receives the new member sequence, with the new member added, it sends back the RS-Ack bit changed, in an MFAS multiframe, as an indication that now it has knowledge of the new sequence. The time needed for this operation is the time of transmitting the MFAS multiframe plus the time of its propagation to the source node ($t_{MFAS} + t_d$). The addition of the new member to the OPUk-Xv is then concluded.

Hence, the overall time delay corresponding to the capacity increase operation is given by:

$$D_{add} = 5t_{MFAS} + t_{MST} + 4t_d. \tag{2}$$

Neglecting the propagation and the node delay the total delay time to add a new member to a VCG is then 64.252 ms for OPU1, 15.995 ms for OPU2 and 3.982 ms for OPU3.

3.3. Connection Capacity Decrease

In an initial state the member to be removed from the OPUk-Xv has its CTRL set to NORM or EOS (if it is the member of the VCG with highest SQ), its SQ set to “i”, and its MST set to OK. In order to decrease the connection capacity the NMS sends a request to the source node for the removal of that member from the VCG. As a result the following operations (see Figure 4) take place:

1) The source node generates and transmits an MFAS multiframe with the CTRL word of the member to be removed changed from NORM or EOS to IDLE. Besides, if the member to be removed is the last member of the VCG, the CTRL word of the previous member is changed from NORM to EOS and its SQ is kept unchanged. If the member to be removed is not the last member of the VCG, the SQ numbers of all following active members are decremented by one and their CTRL words are kept unchanged. This corresponds to the time of generating and transmitting an MFAS multiframe plus the time it takes to reach the destination node ($2t_{MFAS} + t_d$).

2) After receiving CTRL = IDLE the destination sends an MST multiframe with the status of the member to

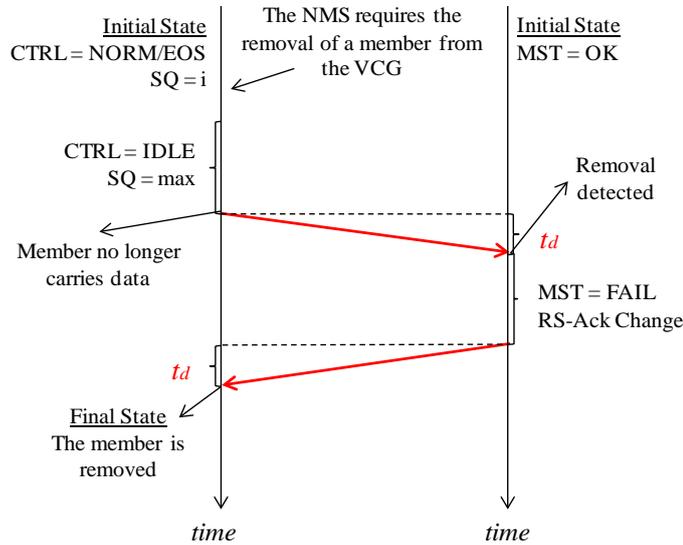


Figure 4. Time diagram for LCAS capacity decrease process.

be removed as FAIL, as an indication of its acknowledgment of the member removal, and the RS-Ack bit changed for that member, as an acknowledgment of the new sequence in the VCG. Thus the delay of this action is the time needed to transmit an MST multiframe plus the time of its propagation to the source node ($t_{MST} + t_d$). The removal of the member from the VCG is then concluded.

Thus, the overall capacity decrease operation delay is given by:

$$D_{rem} = 2t_{MFAS} + t_{MST} + 2t_d. \quad (3)$$

As a consequence, the total time required to remove a VCG member, when the propagation and the node delay are neglected, is 26.641 ms for the OPU1, 6.632 ms for the OPU2 and 1.651 ms for the OPU3.

3.4. Fault Recovery Time

In the presence of a network failure the first step of the recovery process consist in removing the failed member from the VCG. This process involves the following steps (see **Figure 5**):

- 1) The destination node detects a failure in a working VCG member in time instant t_f ($t_f \leq t_d$).
- 2) The destination node removes the failed member from the payload reassembly process and reports the failure to the source by changing its status to MST = FAIL. The delay of this action is then the time needed to transmit an MST multiframe plus the time required to reach the source node ($t_{MST} + t_d$). Note that for a certain period of time the re-assembled payload in the destination side will be harmed, since the traffic is still sent by the source in all the pre-fault members of the VCG.
- 3) When the source node receives MST = FAIL, it notifies the NMS about the detected failure and generates and transmits a new MFAS multiframe with the CTRL word changed to DNU and at the same time stops putting data on the payload area of the failed member. Once the code CTRL = DNU arrives to the destination node, the removal process is complete, and the payload of the VCG is now error free. Thus the delay of this action is the time required to generate and transmit an MFAS multiframe plus its propagation time ($2t_{MST} + t_d$). As a consequence the maximum fault-recovery time can be described as:

$$D_{rec} \leq 2t_{MFAS} + t_{MST} + 4t_d. \quad (4)$$

Neglecting the contribution of the time required to detect the failure ($\leq t_d$), one concludes that the recovery time given by (4) is exactly the same as the time required to the remove a VCG member (3).

In protection schemes based on the “degraded service”, the fault recovery time is given by Equation (4). However, in protection schemes that rely on the existence of protection/backup resources the calculation is different. These schemes require the pre-provision of additional capacity by adding backup members to the VCG in addition to the working members, as a way to protect the last ones. The backup members do not carry any traffic

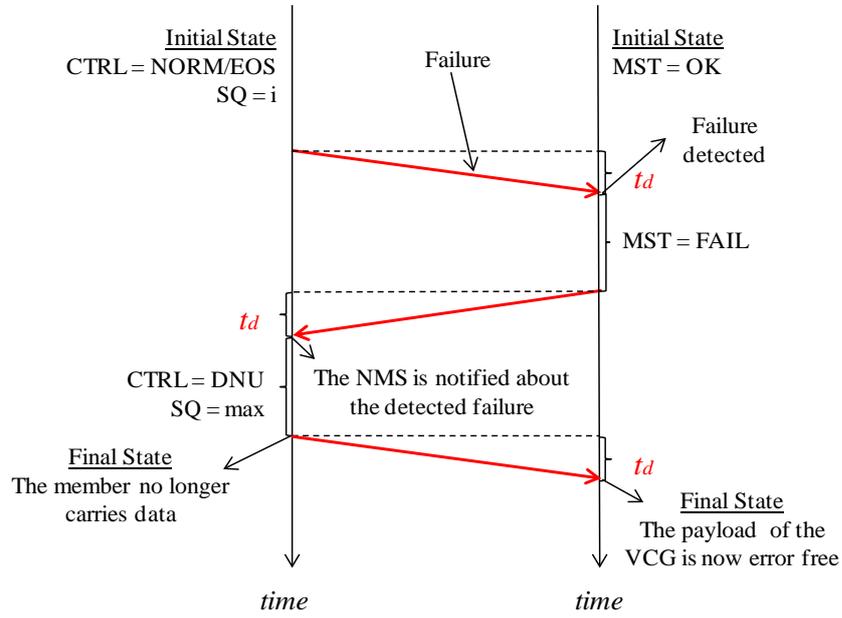


Figure 5. Time diagram for LCAS failed member removal.

during normal operation and to guarantee that they are not used by the destination side in the reassembly process their CTRL status is set to be DNU. Therefore, the fault recovery time in this case, besides the time required for detecting and notifying the failure of a working member, also include the time to activate the backup capacity. The first time includes the steps 1) and 2) of the previously described recovery process, while activating the backup members involves the following actions (see **Figure 6**).

3) After receiving the notification of a member failure the source node notifies the NMS about it and changes the CTRL field of the backup member from DNU to EOS or NORMAL, and the CTRL field of the failed member from EOS or NORMAL to DNU. The SQ numbers are also rearranged. This new coding takes the time $(2t_{MFAS} + t_d)$ to reach the destination side.

4) The destination node detects CTRL = EOS (or NORMAL) for the backup member and consequently will start to transmit MST = OK. Remember that the failed member is already transmitting MST = FAIL. The time taken by this action is $(t_{MST} + t_d)$. Once the source receives MST = OK the traffic previously transmitted on the failed member is switched to the backup member and the recovery process ends. Assuming that the time required by the source to switch the traffic is negligible, the fault recovery time of the described protection scheme reduces to:

$$D_{pro} \leq 2t_{MFAS} + 2t_{MST} + 4t_d \tag{5}$$

when the propagation and node delay are not taken into account the fault recovery time is 28.208 ms for the OPU1, 7.022 ms for the OPU2 and 1.748 ms for OPU3.

The ITU-T Recommendation G.841 [17] indicates that in NG-SDH/SONET networks based on ring protection, for a ring with a perimeter of less than 1,200 km, the switching completing time for a single failure must be less than 50 ms. For a distance of 1,200 km, the propagation time is 6 ms. In this case, neglecting the node latency, we conclude that the worst case fault recovery time is 52.208 ms for the OPU1, 31.022 ms for the OPU2 and 25.748 ms for OPU3, showing that for the OPU2 and OPU3 the values are well below the requisite of the typical value of 50 ms.

4. Simulation Results

The methodology presented in the previous section is used here to evaluate the time delays of the LCAS operations in different OTN networks. For sake of comparison, the time delays for NG-SDH/SONET networks are also evaluated using the results presented in [12]. To obtain the propagation delays the shortest-path between each source-destination node pair was computed using the Dijkstra algorithm.

In our analysis, we considered three network topologies: **Figure 7(a)** the 24-node North American backbone network (UBN), which has 42 bidirectional links and all links are shorter than 3,000 km, **Figure 7(b)** the 19-node European Optical Network (EON) with 36 bidirectional links, and the longest link is about 2,000 km, and **Figure 7(c)** the Pan-European test network (COST 239), which comprises 11 nodes and 26 bidirectional links, and all links are shorter than 1,000 km (the number on each link represents the length in km). In our study it was computed the maximum and the mean delay related to LCAS operations in each network for both NG-SDH/SONET and OTN technologies. The maximum LCAS delays were computed using the shortest path between the two farthest network nodes, while the mean LCAS delays require the knowledge of the mean value of the shortest-paths computed between all network node pairs.

The results obtained for the time delay introduced by LCAS are shown in **Figure 8(a)** for the UBN network, **Figure 8(b)** for the EON network and **Figure 8(c)** for COST 239 network. It was considered the scenarios where connection capacity increases, connection capacity decreases, and a protection switching is performed between a failed working member and a backup member. For the NG-SDH/SONET networks both the Low Order VCAT

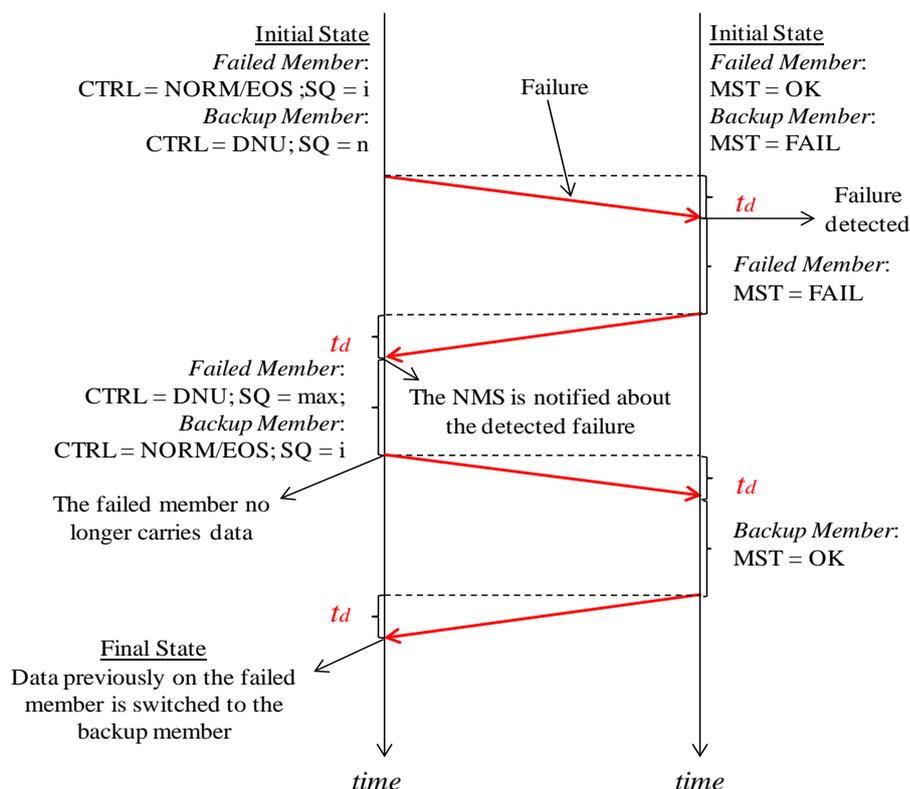


Figure 6. Time diagram for LCAS traffic switching from failed member to backup member.

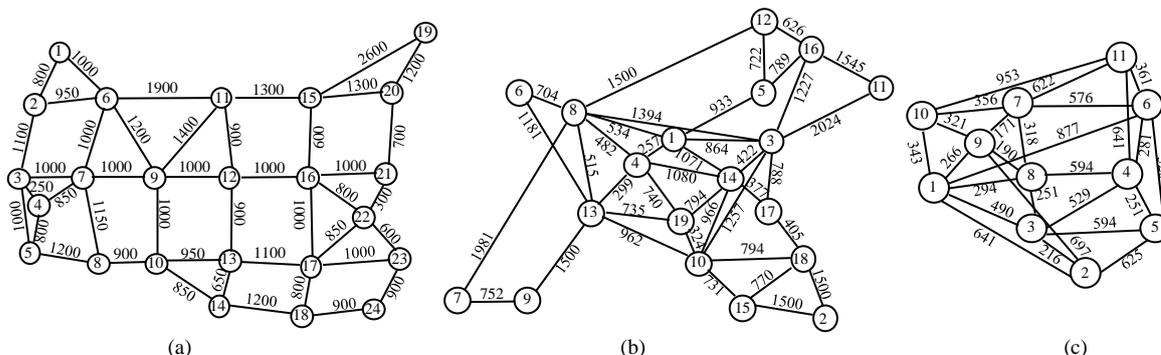


Figure 7. Physical topology of (a) UBN; (b) EON and (c) COST 239 network.

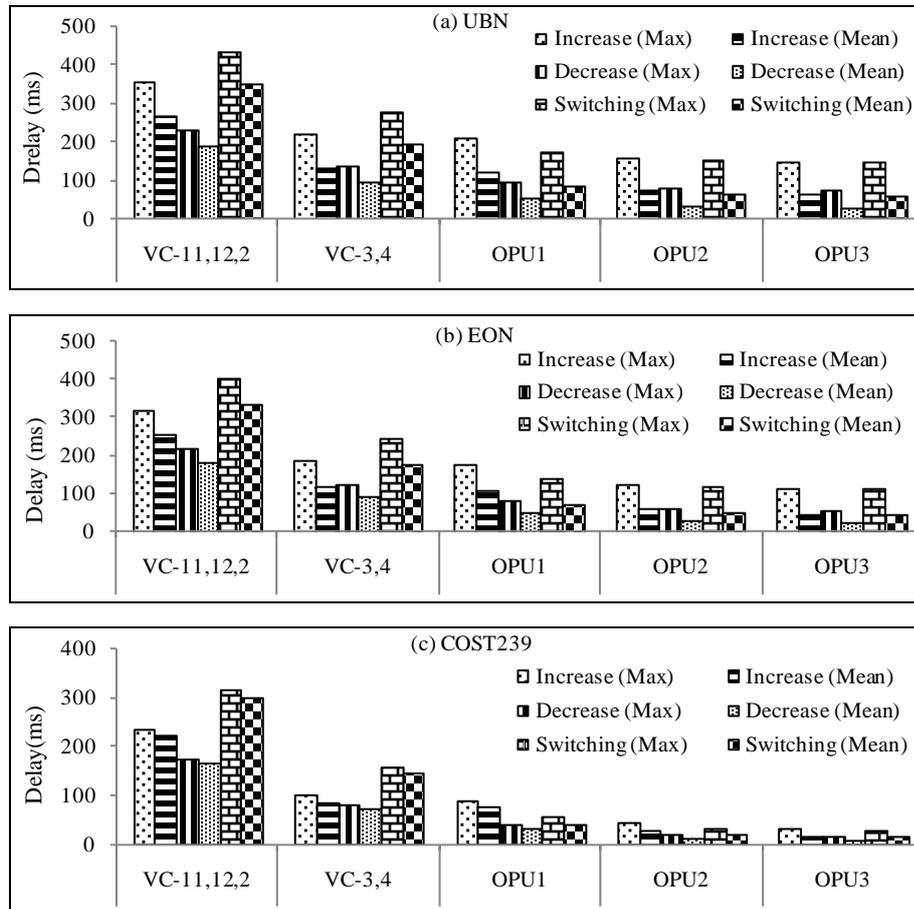


Figure 8. LCAS delays in (a) UBN; (b) EON and (c) COST 239 network.

(LO VCAT) for VC-11, VC-12 and VC-2 concatenated containers, and High Order VCAT (HO VCAT), for VC-3 and VC-4 concatenated containers [18] are considered. Table 3 emphasizes the maximum LCAS time delays for capacity increase and protection switching, since they are the most critical ones.

The most obvious result is the significant difference between time delays in NG-SDH/SONET networks for LO VCAT and HO VCAT signals, being the latter greatly smaller. This difference comes from the fact that the duration of LO VCAT and HO VCAT frames is 500 μ s and 125 μ s, respectively [18], which impacts the duration of the multiframes used in the LCAS analysis.

In OTN networks this delay is even more reduced, since the frame durations are smaller than in NG-SDH/SONET networks. Thus, this becomes immediately an advantage of OTN technology.

In OTN networks the impact of the propagation delay of the messages exchanged between the source and destination nodes is more relevant than it is in NG-SDH/SONET networks, since the OPU_k frames duration is significantly smaller than the VC-*n* frames duration. With $k = 1$ the delay of increasing or decreasing an OPU1-*Xv* connection capacity is similar to the delay of increasing or decreasing, respectively, the capacity of an HO VC-*n*-*Xv*¹ connection. However, for higher values of *k* the delay of LCAS operations decrease substantially, since the multiframe duration also decreases. Regarding to the operation of switching traffic from a failed member to a previously provisioned backup one, the fact of using an OPU1-*Xv* instead of an HO VC-*n*-*Xv* leads to a delay reduction of approximately half.

In all the analyzed scenarios the operation of increasing the link capacity takes more time than the inverse operation, because the handshaking procedure between the source and the destination nodes requires more steps. Furthermore, the difference between the time delays corresponding to these two operations is larger in OTN networks. This is due to the fact that the duration of the MFAS multiframe impacts more the first operation than

¹Notation of a VCAT VCG, where *n* identifies the virtually concatenated containers type.

Table 3. Maximum LCAS time delay for capacity increase and protection switching (ms).

| Technology | Container | Capacity Increase Delay (ms) | | | Protection Switching Delay (ms) | | |
|------------|-----------|------------------------------|-----|----------|---------------------------------|-----|----------|
| | | UBN | EON | COST 239 | UBN | EON | COST 239 |
| NG-SDH | LO VCAT | 352 | 317 | 236 | 432 | 397 | 316 |
| | HO VCAT | 218 | 183 | 102 | 276 | 241 | 160 |
| | OPU1 | 208 | 173 | 92 | 172 | 137 | 56 |
| OTN | OPU2 | 160 | 125 | 44 | 151 | 116 | 35 |
| | OPU3 | 148 | 113 | 32 | 146 | 111 | 29 |

the second one (see Equations (2) and (3)) and this duration is much larger than the duration of the MST multiframe, contrary to what happens in the NG-SDH/SONET networks, where the MST multiframe is longer than the multiframe used for the MFI control [18]. For the same reason, the operation of protection switching from one member to another is quite large in NG-SDH/SONET since the destination node transmits two MST multiframes to the source node. As for OTN, this operation's delay is similar to the capacity increase delay, since the propagation time has a major impact in the overall operation delay and the number of multiframes exchanged in both cases is equal. Note, for example, that in COST 239 network we can accomplish with an OPU3-Xv a delay of 29 ms, while in a NG-SDH/SONET networks we get no less than 160 ms.

The multiframe delays are constant for each LCAS operation and each VC-n-Xv or OPUk-Xv. Therefore, delays suffered during the process of dynamically allocating or freeing bandwidth vary with the distance between network nodes. Hence, LCAS operation delays are network topology dependent.

The UBN network, which presents longer path distances, naturally suffers bigger delays and, as a consequence, the contribution of the propagation delay to the LCAS delay is stronger than, for example, in the COST 239 network. The maximum distance between two nodes in the latter network is 1,386 km. This leads to a maximum propagation delay around 36 ms, for increasing the connection capacity. For the UBN network the maximum distance between network nodes goes up to 7,200 km leading to a maximum propagation delay of about 144 ms.

5. Conclusion

In this paper, we have explored the application of VCAT/LCAS techniques to provide dynamism in the context of OTN networks. A detailed explanation about the procedures used to resize the capacity of the connections is presented and the time-delays associated with the process are computed. A comparison with NG-SDH/SONET networks is also provided. It is shown that the resizing operations in OTN networks are faster than in NG-SDH/SONET networks and the speed of the process increases when we move from OPU1-Xv connections to OPU3-Xv connections. For example, for the first type of connection the maximum time delay obtained in all the reference networks considered was about 200 ms, while for the second one the maximum delay is reduced to about 150 ms. For the sake of comparison, the worst results for NG-SDH/SONET networks were about 220 ms and 350 ms, for HO-VCAT and LO-VCAT, respectively. Our results have also highlighted the interest of applying the VCAT/LCAS techniques as a way to improve resilience: by adding a backup member to protect a working member in a VCG we showed that using OPU3-Xv connections it is possible to recover from a member failure in about 25 ms, in a scenario where the NG-SDH/SONET standards define a maximum value of 50 ms.

References

- [1] Gumaste, A. and Krishnaswamy, N. (2010) Proliferation of the Optical Transport Network: A Use Case Based Study. *IEEE Communications Magazine*, **48**, 54-61. <http://dx.doi.org/10.1109/MCOM.2010.5560587>
- [2] Carrol, M., Roesse, J. and Ohara, T. (2010) The Operator's View of OTN Evolution. *IEEE Communications Magazine*, **48**, 46-52. <http://dx.doi.org/10.1109/MCOM.2010.5560586>
- [3] ITU-T Rec. G.709 (2009) Interfaces for the Optical Transport Network (OTN).
- [4] ITU-T Rec. G.7044/Y.1347 (2011) Hitless Adjustment of ODUflex (GFP).

- [5] ITU-T Rec. G.7042/Y.1305 (2006) Link Capacity Adjustment Scheme (LCAS) for Virtual Concatenated Signals.
- [6] Bernstein, G., Caviglia, D., Rabbat, R. and Van Helvoort, H. (2006) VCAT/LCAS in a Clamshell. *IEEE Communications Magazine*, **44**, 34-36. <http://dx.doi.org/10.1109/MCOM.2006.1637944>
- [7] Santos, J., Pedro, J., Monteiro, P. and Pires, J. (2011) Optimized Routing and Buffer Design for Optical Transport Networks Based on Virtual Concatenation. *Journal of Optical Communications and Networking*, **3**, 725-738. <http://dx.doi.org/10.1364/JOCN.3.000725>
- [8] Acharya, S., Gupta, B., Risbood, P. and Srivastava, A. (2004) PESO: Low Overhead Protection for Ethernet over SONET Transport. *IEEE Infocom 2004*, Hong Kong. <http://dx.doi.org/10.1109/INFCOM.2004.1354491>
- [9] Roy, R. and Mukherjee B. (2008) Degraded-Service-Aware Multipath Provisioning in Telecom Mesh Networks. *OFC/NFOEC 2008*, San Diego, 24-28 February 2008. <http://dx.doi.org/10.1109/OFC.2008.4528661>
- [10] Huang, S., Martel, C. and Mukherjee, B. (2011) Survivable Multipath Provisioning with Differential Delay Constraint in Telecom Mesh Networks. *IEEE/ACM Transactions on Networking*, **19**, 657-669. <http://dx.doi.org/10.1109/TNET.2010.2082560>
- [11] Ou, C., Sahasrabudhe, L., Zhu, K., Martel, C. and Mukherjee, B. (2006) Survivable Virtual Concatenation for Data over SONET/SDH in Optical Transport Networks. *IEEE/ACM Transactions on Networking*, **14**, 218-231. <http://dx.doi.org/10.1109/TNET.2005.863462>
- [12] Han, D., Li, X. and Gu, W. (2006) The Impact of LCAS Dynamic Bandwidth Adjustment on SDH/SONET Network. *Journal of Optical Communications*, **27**, 317-320. <http://dx.doi.org/10.1515/JOC.2006.27.6.317>
- [13] Pedro, J., Santos, J. and Pires, J. (2011) Performance Evaluation of Integrated OTN/DWDM Networks with Single-Stage Multiplexing of Optical Channel Data Units. *ICTON 2011*, Stockholm, 26-30 June 2011. <http://dx.doi.org/10.1109/ICTON.2011.5970940>
- [14] Schulzrinne, H., Casner, S., Frederick, R. and Jacobson, V. (2003) RTP: A Transport Protocol for Real-Time Applications. *RFC 3550*.
- [15] Bernstein, G., Caviglia, D., Rabbat, R. and van Helvoort, H. (2011) Operating Virtual Concatenation (VCAT) and the Link Capacity Adjustment Scheme (LCAS) with Generalized Multi-Protocol Label Label Switching (GMPLS). *RFC 6344*.
- [16] Das, S., Parulkar, G., Singh, P., Getachew, D., Ong, L. and McKeown, N. (2010) Packet and Circuit Network Convergence with Open Flow. *OFC/NFOEC'10*, San Diego, 24-28 March 2010.
- [17] ITU-T Rec. G.841 (1998) Types and Characteristics of NG-SDH Network Protection Architectures.
- [18] Van Helvoort, H. (2005) Next Generation NG-SDH/SONET: Evolution or Revolution? John Wiley & Sons, Ltd., Chichester. <http://dx.doi.org/10.1002/0470091223>

High-Level Portable Programming Language for Optimized Memory Use of Network Processors

Yasusi Kanada

Central Research Laboratory, Hitachi, Ltd., Yokohama, Japan
Email: Yasusi.Kanada.yg@hitachi.com

Received 26 January 2015; accepted 15 February 2015; published 16 February 2015

Copyright © 2015 by author and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Network processors (NPs) are widely used for programmable and high-performance networks; however, the programs for NPs are less portable, the number of NP program developers is small, and the development cost is high. To solve these problems, this paper proposes an open, high-level, and portable programming language called “Phonepl”, which is independent from vendor-specific proprietary hardware and software but can be translated into an NP program with high performance especially in the memory use. A common NP hardware feature is that a whole packet is stored in DRAM, but the header is cached in SRAM. Phonepl has a hardware-independent abstraction of this feature so that it allows programmers mostly unconscious of this hardware feature. To implement the abstraction, four representations of packet data type that cover all the packet operations (including substring, concatenation, input, and output) are introduced. Phonepl have been implemented on Octeon NPs used in plug-ins for a network-virtualization environment called the VNode Infrastructure, and several packet-handling programs were evaluated. As for the evaluation result, the conversion throughput is close to the wire rate, *i.e.*, 10 Gbps, and no packet loss (by cache miss) occurs when the packet size is 256 bytes or larger.

Keywords

Network Processors, Portability, High-Level Language, Hardware Independence, Memory Usage, DRAM, SRAM, Network Virtualization

1. Introduction

To enable programmability for networking and in-network processing, especially for new network-layer pro-

programming for clean-slate virtual-networks [1], network processors (NPs) have been used [2] and will be more widely used in the near future. NPs, which were developed for software-based high-performance networking solutions, make it possible to quickly develop arbitrary protocol and functions in the case of hardware-based solutions as well.

However, there are three problems that make using NPs for such functions difficult. The first problem is lack of portability. Because low-level languages that are similar to assembly languages must be used for developing NP programs, the programs are not portable. Although extended versions of C can usually be used for developing NP programs, essential libraries depend on vendor-specific proprietary hardware and software, and proprietary rights on NP programs are protected by non-disclosure agreements (NDAs) preventing programs and documents concerning an NP being ported. The second problem is high development cost and that the availability of NP program developers is limited. NP program developments require special skills, and the knowledge they require is not widely available; thus, only a limited number of developers have the ability to develop NP programs. In addition, vendor-specific information is required in NP-program development. Consequently, the learning curve of NP-program development is very gentle, the development takes a very long time, and its cost is very high. The third problem is restriction on publishing developed programs, papers, and documents concerning an NP. This is a serious problem for network researchers.

The three above-described problems can be solved by successfully designing and implementing a high-level language, which can translate programs into NP machine code or a vendor-dependent C program. Programs written in this language must be translated into NP-dependent object programs; however, to solve the problems, the language must be hardware- and vendor-independent.

An important common NP feature concerning high-performance packet processing (to avoid packet drops caused by cache misses) is to use static random-access memory (SRAM) and dynamic random-access memory (DRAM) by different methods with explicit awareness by the programmer, making programming difficult and time-consuming. Although memory allocation is not the only issue that causes the above three problems, this is the most important and serious issue because NPs are optimized for wire-rate processing and memory abuse immediately prevents it and severely reduces the performance. In particular, whole packets are stored in DRAM, and only the headers, which must be modified, removed, or added, are cached in SRAM because if data stored in DRAM is accessed by a CPU core, access takes an excessively long time, and wire-rate processing is impossible. The rest of the packets are just forwarded to the next network node without modification in the NP. This is common because it is necessary for NPs to store packets in memory while processing them, but the size of SRAM is limited, so whole packets cannot be stored in short-access-time memory, *i.e.*, SRAM.

When programming a packet-processing program for NPs, programmers must use an assembly language or C with assembly-level features, and must be very careful to get high performance. When using general-purpose CPUs, programmers can use high-level language and do not have to distinguish SRAM (or cache) and DRAM because they are automatically selected when programs load and store data. However, NP programmers must usually know whether the packet to be processed is on SRAM or DRAM (or both) because this knowledge is critical for attaining stable (*i.e.*, mishit-less) wire-rate processing. Two types of NP architectures are available. In one of them, such as Intel IXP, the SRAM and the DRAM are different classes of memory with different addresses. In the other type, such as Cavium Octeon[®], the SRAM can be accessed as cache or registers, in a similar manner to general-purpose CPUs, but programmers must still be aware of the SRAM/DRAM distinction because the NP handles them in different ways. These cases are explained in more detail in Section 2.

Although it is a promising approach to design a new open and portable high-level language and to implement a high-performance language processor, *i.e.*, a compiler and run-time routines, it is still very hard to solve the above three problems because of the wide semantic gap between the language and the object program.

However, this paper describes the successful first step toward this goal. Hardware features such as those described above can be abstracted to common high-level language features that do not make programmers conscious of the low-level hardware features. To enable this type of abstraction, a high-level language called “Phonepl” (portable high-level open network processing language) is proposed, and a method for compiling packet-handling programs in Phonepl into high-performance programs that can fully utilize hardware while distinguishing SRAM and DRAM is proposed. Here, “open” means that network processors can be programmed without NDAs. Especially, packet headers are automatically cached, the language processor is aware that the data being handled is stored in either SRAM or DRAM (or both) and manages data transmission between them, and programmers do not have to pay attention to this distinction, so the programming cost can be decreased.

Phonepl does not depend on vendor-specific NP hardware and software, and thus the programs in Phonepl can be portable among various NPs.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 describes Phonepl. Section 4 describes a method for implementing Phonepl for NPs, especially four representations of packet type and a method for handling them. Section 5 describes a prototype implementation of Phonepl for plug-ins for a network-virtualization environment called the VNode Infrastructure, and Section 6 evaluates it by using several applications. Section 7 concludes this paper.

2. Related Work

This section focuses on previous studies on NPs and languages for packet processing because, although there are many studies on memory-related optimizations concerning high-performance computing, such as Sequoia [3], they focus on array processing and the requirements for packet-stream processing are quite different from them.

2.1. Selection of SRAM/DRAM in NPs

The IXP series of NPs developed by Intel [4] does not have cache, and its SRAM and DRAM have different memory spaces. The developers at Intel reported that cache is not effective in the case of NPs, so this type of memory architecture is good for network processing. However, it is difficult to program IXP processors because programmers, who are not even aware of the difference between SRAM and DRAM, must use them with different methods.

In contrast, the architectures of NPs developed later, for example, Cavium Octeon[®] [5] and Tilera[®] Tile Processors [6], are more similar to those of general-purpose CPUs with cache. However, because a cache miss may disable wire-rate transmission of packets, there are several devices that can be applied to avoid cache miss. That is, data to be processed at wire rate must be stored in SRAM. However, because an NP cannot usually have sufficient quantity of SRAM to store all the processing packets, it only stores descriptors and headers of packets in SRAM, and the rest or whole packets must be stored in DRAM. Various different types of packet-processing hardware and software behave in a similar way. In addition, to process packets at wire rate, NPs distribute packets to many cores for parallel processing, and they sort the resulting packets by hardware in input order and queue them for output or the next processing.

2.2. Selection of SRAM/DRAM with a Packet-Processing Language

In an NP program-development environment called Shangri-La [7], which was developed by Intel and several universities, a high-level language called Baker [4] was developed for IXP. By assuming that packet bodies are stored in DRAM and descriptors are stored in SRAM, Baker enabled programmers to handle packet data without having to consider whether they are on DRAM or SRAM. The data structure on SRAM, however, must be designed by programmers, so it depends on NP architecture. In addition, programmers must describe data transmission between DRAM and SRAM, so they must explicitly describe caching operations.

Unlike Octeon or Tilera, Baker does not have a mechanism for supporting automatic distinguished use of SRAM and DRAM. It is therefore difficult to process packets at wire rate by using Baker.

2.3. Packet-Stream and Data-Stream Languages

Click [8] is software architecture for describing routers modularly. Two-level description is used in Click. The lower level, or component level, is described by C, and the higher level is described by a domain-specific language, which connects modules in several ways. Click programs can be portable, but it is difficult to get high performance from portable Click programs. NP-Click [2] is a specialized implementation of Click for IXP NPs. Modules in NP-Click are written in IXP-specific C language; therefore, the programs are not portable.

Frenetic [9] is a language for controlling a collection of OpenFlow [10] switches. It is embedded in Python but is based on SQL. It is a declarative language and processes collections (streams) of packets instead of processing individual packets procedurally in the manner of Phonepl. Because Frenetic processes packet streams, it is very similar to CQL (Continuous Query Language) [11]. Unlike Phonepl, Frenetic can only be used to program the control plane; it cannot handle the data plane.

NetCore [12] is a rule-based language for controlling OpenFlow switches. Rules in NetCore are condition-

action rules; that is, rules that match incoming packets are activated.

3. Packet-Processing Language

A high-level language called Phonepl, which solves the three problems described in the introduction, is outlined.

3.1. Basic Design of Phonepl

Phonepl is designed for wire-rate (low-level) packet-processing of any format, such as a non-IP and/or non-Ethernet format, as well as designed to be as close as a conventional programming language, *i.e.*, Java, because it should be easy to handle by Java and C++ programmers.

The reason why a new language, which is open, portable, and easy to use, is designed is explained as follows. Although it is close to conventional languages, a new language is required because it is very hard to compile a general-purpose program to high-performance object program for NPs, which very optimized hardware-usage, especially memory usage, is required for. Phonepl may thus be considered as a very restricted and extended version of Java.

Two major design goals of Phonepl are as follows. First, Phonepl must be high-level; that is, it must be designed for the programmer not to be aware of proprietary hardware and software. Second, Phonepl must be able to express high-performance packet-processing programs. Especially, processing at wire-rate, *i.e.*, 10 Gbps or more, without packet drops is required. In an NP, input packets may be partially cached, that is, the header of the packet is stored in SRAM and the rest or whole packet is stored in DRAM, but a DRAM access may disable wire-rate processing and cache miss easily cause packet-drops. However, this goal must be achieved without abandoning the first goal, *i.e.*, high-level programmability.

To achieve these design goals, data structures, especially Packet and String, which are the most important data structures in Phonepl, must be carefully designed and the method for processing them must be developed. Especially, packets are designed to be immutable byte strings in Phonepl and they are distinguished from non-packet strings.

There are five language features concerning this design. The first feature is that packets are byte strings because packets with arbitrary formats should be able to be handled in uniform methods. Packets have variable length, so they can be handled as byte strings (similar to character strings). A packet in Phonepl is not a encapsulated object. This decision makes low-level and cross-layer optimization of packets easier. The protocol-handling method written in Phonepl is thus completely different from that written in Java.

The second feature is that packets are immutable. Packets are handled as immutable (non-rewritable) objects, which are similar to character strings in Java or other languages; that is, packet contents cannot be rewritten. This immutability enables memory areas, especially DRAM areas, to be shared by packets before and after an operation.

The third feature is that types of packets, *i.e.*, Packet, and non-packet strings, *i.e.*, String, are different in Phonepl. They are incompatible for two reasons. First, although they can be logically identical, they must be implemented by using quite different methods and this distinction makes implementation more efficient and easier. Operations such as subpacket and substring described below utilize this difference. Second, programmers can easily distinguish them. Non-packet strings are used for temporary data, *e.g.*, packet fragments, but packets are used for I/O data; that is, packets and packet fragments (non-packets) are different for programmers.

Two assumptions are made in regard to implementation of these data types. The first assumption is that whole String objects are stored in cacheable memory, *i.e.*, in SRAM, but can be stored in DRAM if needed. If they are in cache, purging the cache may have to be inhibited. The second assumption is that only the head of a packet is cached, and the tail is stored only in DRAM. However, a short packet may be wholly cached and may be stored only in SRAM.

The fourth feature is that packet and non-packet byte-substring operations are different in Phonepl because the types of the operation results are different. A new packet can be generated by removing part of another packet using a subpacket operation, and a non-packet byte string can be generated by extracting part of a packet using a substring operation. These operations can have the same name *i.e.*, a substring, but are distinguished.

The fifth feature is that packet- and byte-*concatenation* operations are specialized. A byte string can be generated by concatenating two or more byte strings by a concat operation, and a packet can be generated by concatenating one or more byte strings and a packet by a packet constructor called “new Packet”. Although a packet

can logically be generated by concatenating multiple packets, such concatenation seems to be practically less useful and difficult to implement, so no such operation is included (See Section 4.2.3 for more explanations).

3.2. Program Example and Packet Operations

To outline Phonepl and to explain several data structures and important packet operations, a program that performs MAC-header addition/removal, which cannot be performed by conventional non-programmable network nodes, is shown in **Figure 1**. The program in this figure defines class `AddRemMAC`. It has two functions that handle two bidirectional packet streams, *i.e.*, `NetStream1` and `NetStream2` (lines 001 - 002), which are bound to physical network interfaces outside this program. One function inputs packets from `NetStream1`, generates new packets with a new MAC header (*i.e.*, adds a new MAC header at the front) for each packet, and outputs them to `NetStream2`. The other function inputs packets from `NetStream2`, removes the MAC header in front, and outputs it to `NetStream1`. The program is much simplified because it is sufficient to show the functionality and basic implementation of the language; that is, no validation test is performed before the header is added or removed. However, it is easy to add check code to this program.

Packet flows are handled as “streams” in Phonepl. Method of stream handling is described using the constructor of class `AddRemMAC` here. The parameter declarations of `AddRemMAC` (lines 006 - 007) specify that input packets to parameter `port1` pass to method `process1` and input packets to parameter `port2` pass to method `process2`. This type of parameter declaration is Phonepl specific; that is, Java grammar is modified for the sake of stream processing. The parameter values (packet streams) are assigned to instance variables `out1` and `out2` to make them available in the newly created object. Methods `process1` and `process2` receive one packet at a time. (One of these methods is executed once on only one core for each packet.) Because Phonepl handles input packets by these methods only, there is no specific method or statement for packet input.

Examples of a substring operation (which is used for accessing packet components), a packet constructor (which is used for packet composition), and a packet-stream output using “put” method can be seen in method

```

001 import NetStream1;
002 import NetStream2;

003 class AddRemMAC {
004     NetStream out1;
005     NetStream out2;

006     public AddRemMAC(NetStream port1 > process1,
007                     NetStream port2 > process2 ){
008         out1 = port1;
009         out2 = port2;
010     }

011     void process1(Packet i) {
012         //Port 1 to 2 (no VLAN -> no VLAN)
013         Packet o = new Packet(i.substring(0,14),i);
014         // MAC header of original packet (i: Original packet)
015         out2.put(o);
016     }

017     void process2(Packet i) {
018         // Port 2 to 1 (no VLAN -> no VLAN)
019         Packet o = i.subpacket(14);
020         // remove MAC header (no VLAN)
021         out1.put(o);
022     }

023     void main() {
024         new AddRemMAC(new NetStream1(),
025                       new NetStream2());
026     }
027 }

```

Figure 1. Simple MAC-header addition/removal program.

process1 (line 011). This method handles a packet that comes from NetStream1, generates a byte string from the first 14 bytes of input packet *i* (it is assumed that the size of MAC header is 14 bytes) by `i.substring(0,14)`, generates a packet by concatenating this byte string and the original packet by `new Packet(...,i)`, and outputs the resulting packet to NetStream2(out2).

An example of subpacket operation, which generates packets from an existing packet, can be seen in method process2 (line 015). This method handles a packet that comes from NetStream2, generates a packet by removing the first 14 bytes of input packet *i* by `i.subpacket(14)`, and outputs the resulting packet to NetStream1 (out1).

Finally, an example of stream initialization is seen in function main() (line 019). When class AddRemMAC is initialized, this function is executed. It logically runs only once, but each processor core may execute it once unless there are side-effects. It generates an instance (a singleton) of class AddRemMAC, which runs forever and processes packets repeatedly unless it is externally terminated. Two packet streams are generated and passed as arguments of AddRemMAC. They start to operate (input and/or output packets) when instances are generated.

4. Implementation Method

To implement semantics close to conventional programming languages such as Java, a special method of handling data (object) is required for Phonepl. The key feature of Phonepl implementation is the four representations of packets and operations among them.

4.1. Four Representations of Packets

In Phonepl, multiple packet data-representations used in NPs are unified as a single data type called Packet. Four different representations shown in **Figure 2(a)** (explained below) are therefore used for Packet. These representations are required because of the following two reasons concerning high-performance packet-processing and NP hardware. First, in most packet-processing in network nodes, packet headers are added, removed, or updated, but packet tails, *i.e.*, payloads, are not touched unless very deep packet-inspection is required. So the packet headers must be stored in SRAM (or scratchpad memory) but the packet tails can be stored in DRAM as described in the introduction and in the previous section. It is usually not possible to cache whole packet. Second, NPs are designed to handle input and/or output packets by specialized hardware. The hardware is optimized for the packet-processing requirements described above, but some hardware-specific restrictions apply in addition.

An example of hardware-specific data representation that matches the abstract representation is shown here. In some NPs, there are input-specific and output-specific packet formats using a special descriptor format. Short packets may be fully stored in SRAM but packet heads may be stored in both SRAM and DRAM for longer packets. The four abstract representations are designed to generalize various concrete representations, such as shown in **Figure 2(b)**, used in NPs. Although the descriptor format is specialized, it can be abstracted as shown in **Figure 2(a)**. If vendor-specific C language is used, these representations are handled separately; however, Phonepl, handles them uniformly. Even for cases that the NP has a cache, it is probably useful to distinguish multiple representations because cache miss must be avoided.

The four representations are explained in the following.

- **Cached:** The whole packet data is stored in SRAM. It is not assumed that a copy of the data is stored in DRAM.
- **Mixed:** The head of a packet (the number of bytes depends on implementation) is stored in SRAM, and whole packet data is stored in DRAM.
- **Gathered:** A packet consists of multiple fragments. Each fragment is stored in a memory area (*i.e.*, DRAM or SRAM). A gathered packet can be represented by an array or a linked list of fragments.
- **Uncached:** The whole packet is stored in DRAM. It is not assumed that a copy of the data is stored in SRAM.

Packets inputted to NPs are usually in cached or mixed representation; that is, short packets may be represented by cached representation but mixed representation is required for long packets. All four representations are used for expressing operation results and may be used for output. However, reasoning of mixed, gathered, and uncached representations are explained more.

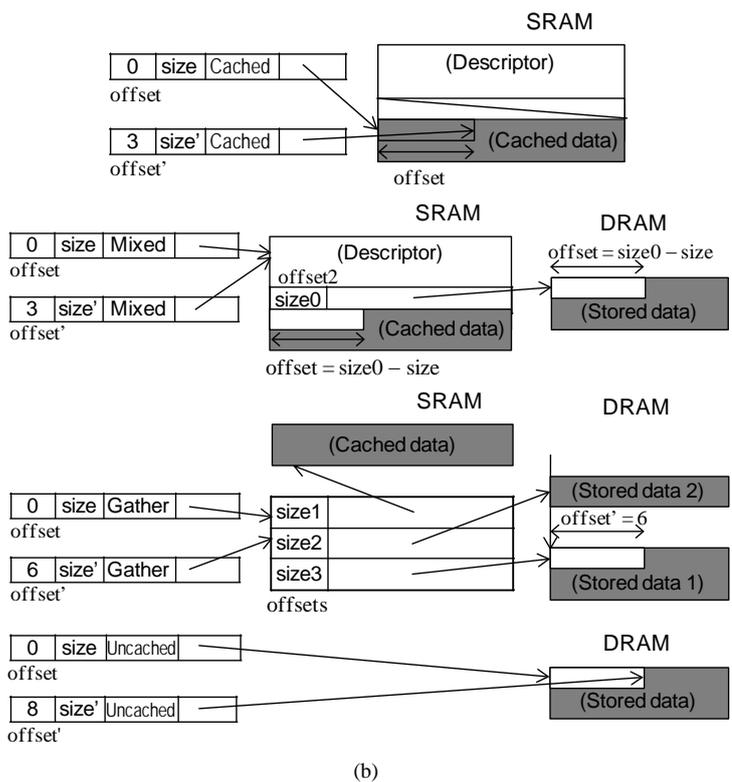
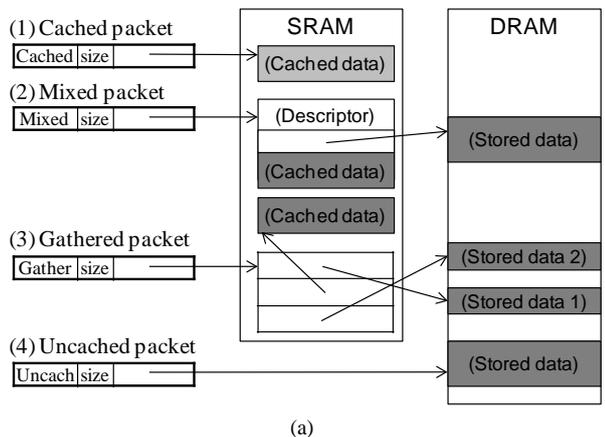


Figure 2. Four representations of packet type. (a) Abstract representations; (b) Examples of more detailed representations.

Mixed representation is required because, in packet processing, only the packet head (containing headers) is usually modified, headers are added or deleted, and the packet tail is kept unchanged. Good performance can therefore be obtained by caching only the head to SRAM and storing the tail only in DRAM. Data accessed by cores must be stored in SRAM because if data stored in DRAM is accessed, it takes excessively long time, and wire-rate processing becomes impossible.

Gathered representation is required when generating a packet from multiple pieces of data stored in DRAM or SRAM. In such a case, if all the pieces are copied to a contiguous area (of DRAM), copy from DRAM to DRAM is required and wire-rate processing becomes impossible. This representation is closely related to the immutability of packets, which enables sharing part of a string.

Uncached representation is required when a packet is generated from a tail of another packet with gathered representation by an operation such as a header deletion.

Because the four representations may have to be distinguished at run time, a tag must be supplied. The tags should be in packet-data pointers. However, because packet data are handled by hardware in NPs, the data representation and handling methods in the case of a high-level language must be very carefully designed and implemented. If the address space is sufficiently large, a part of the address can be used for a tag. This representation is close to widely used methods for dynamically-typed languages, such as Python or Lisp.

4.2. Packet Operations and Four Representations

Because there are four packet-data representations and each packet data has a tag, packet operations must be implemented for all these representations, and sometimes run-time tag check is required.

4.2.1. Run-Time Tag Check

Because there are multiple representations in Packet type, they must be distinguished dynamically (by the run-time routines in the NP) or statically (by the Phonepl compiler). In terms of efficiency, it is better for the representation to be statically distinguished. However, it is impossible to distinguish every representation of a packet statically, so run-time tag-check is, at least sometimes, necessary. Especially, if a non-optimizing compiler is used, tag check is always necessary at run time. Such a run-time check causes overhead, but it does not usually prevent wire-rate processing because the tags are in cached pointers and a tag can be added and removed with very small cost.

4.2.2. Packet I/O

Some NP hardware creates a descriptor when receiving a packet. The descriptor is in SRAM, and whole packet data may be stored in DRAM. The input packet format, thus, is close to mixed representation (or cached representation in the case of a short packet); however, a tag must be added when run-time tag-check is required. The run-time routine should thus decide which representation is to be used and insert the tag value. This means that the language processor must fill the gap (*i.e.*, convert) between data representations in the hardware and in Phonepl. If the gap is wide, significant CPU time is required to fill it, and performance may decrease. An appropriate representation design is therefore important.

An output packet format must be prepared for some NPs when sending a packet. One of the four representations should be close to the output format; however, the tag must be removed before passing the data to the packet output hardware. For example, the output format may be close to gathered representation, but the tag value “gathered” must be cleared. The hardware concatenates the fragments pointed to by the gathered representation and outputs the result.

4.2.3. Subpacket

Each representation requires different implementations of an operation to achieve a subpacket operation. In all the cases described below, the operations are executed using data stored in SRAM, and DRAM is not accessed.

If the packet has a cached representation, a subpacket of the packet is in a cached format. The original packet can be stored in the allocated SRAM area. The resulting subpacket may share the original packet data or may be a copy of the original data. In this case, because both the original and copied data are stored in SRAM, this copy operation probably does not prevent wire-rate processing.

If the packet has a mixed representation, a subpacket of the packet may be in a mixed or uncached format. That is, there are two cases. Firstly, if the resulting packet contains both head data stored in SRAM and tail data stored in DRAM, the result is mixed format. Secondly, if the resulting packet only contains tail data, the result is uncached format. In general, the resulting representation is not known at compile time because the range specified in subpacket operation might not be known at compile time. In both cases, a new descriptor is generated in SRAM by using the original descriptor, but no packet data stored in DRAM is accessed.

If the packet has a gathered representation, a substring of the packet is usually in a gathered format. The original and resulting packets may share the array of fragments (*i.e.*, only a packet-type pointer is generated) or the resulting pointer may point to a new array copied from the original array. An array copy probably does not prevent wire-rate processing because both arrays are stored in SRAM.

If the packet has an uncached representation, a substring of the packet is in an uncached format. Both the original and resulting packet data are stored in DRAM and shared. The address and the length of the resulting

packet are stored in a packet-type pointer. No packet data stored in DRAM are accessed.

4.2.4. Concatenation

When a packet is generated by concatenating one or more byte strings (such as new headers and a packet content), a constructor, “new Packet()”, is used. In the current implementation method, this constructor generates a gathered-format packet. That means, the parameter values of the constructors are the elements of the array in the gathered format. However, a more optimized method, which uses other representations, may be developed.

The last element of the constructor may be a packet of any representation. If this element has a mixed format, the DRAM part (which represents the whole packet) becomes an element of the array. If this element has a gathered format, each input array element becomes an element of the array of the output gathered format.

4.2.5. Generating Packet without Using Input Packet

A packet can be created without using a pre-existing packet by using a packet constructor. The generated packet is in cached or gathered format. If the constructor has only one argument that contains a byte string, the resulting packet is in cached format, and if it has two or more arguments, the resulting packet is in gathered format.

4.3. Several Miscellaneous Issues

Two issues related to the proposed packet-handling method are explained in the following. The first issue is memory deallocation. Sharing part of packets and strings makes memory deallocation difficult. Garbage collection or reference counting can solve this problem completely, but the overhead is large. In the current implementation, strings that are (potentially) assigned to global (instance) variables are not deallocated. However, the current deallocation policy may cause memory leak. A more precise method should be devised in future work.

The second issue is adaptation to hardware-based memory allocation. Some NPs allocate and deallocate packet memory automatically to avoid software-memory-management overhead. When a packet arrives, the SRAM and DRAM required for the packet is allocated. However, it is difficult for NP hardware to decide when the packet memory can be deallocated. A Phonepl compiler must therefore generate code for deallocate it.

5. Prototyping

The above-described implementation method has been applied to a programming environment called +Net, which contains a Phonepl processor called +Net Phonepl. +Net Phonepl is used for programming physical nodes with a network-virtualization function and NPs.

5.1. Platform

The prototype compiles a Phonepl program and runs it on a “virtualization node” (VNode) [1] [13]. A virtualization platform called VNode Infrastructure, which supports multiple slices (*i.e.*, virtual networks) using a single network infrastructure, and a high-performance fully functional virtualization testbed were developed. The components of a VNode contain NPs. The prototype is a replacement of one or more NPs in this environment. The program has packet I/O streams as described in Section 3.

A source program is compiled according to the following procedure. First, an intermediate language program (ILP) is generated by using a Phonepl syntax/token analyzer. The syntax analyzer was generated using “Yet Another Perl Parser” (YAPP) compiler, which has similar functions as those of YACC (Yet Another Compiler Compiler) or Bison parser-generators but is written in and generates Perl code. The ILP is translated by using a Phonepl translator into a specialized C program. A GNU C compiler for Oocteon compiles this C program and generates object code for an Oocteon board called WANic-56512 developed by General Electric Company. A run-time library is linked to the object program. The main components of this library are an initializer, packet processors, and a packet-output routine.

5.2. Compiled Code of +Net-Phonepl Compiler

To outline the object-code structure and the compilation (or program transformation), an example of compiled code is explained here. The C program generated by the Phonepl compiler from the MAC-header addition/removal program (in Figure 1) is shown in Figure 3.

```

// Translated Code for Octeon 58XX (WANic 56582) by Phonpl Translator
#include <stdio.h>
#include <string.h>
#include "runtime.h"
#include "cvmx-helper.h"

// Packet handler vector:
void (*__packetHandler[17])(__Packetp p);

// Stream data type for packets:
typedef int NetStream;

// Method NetStream.put(uint64_t port, Packet outp)
extern int NetStream_put(uint64_t port, __Packetp outp);

// Omitted

// Instance variables of the singleton instance (Singleton assumed!)
typedef struct
{
    NetStream out1;           (1) Derived from instance variable
    NetStream out2;         (out1, out2) declaration
} AddRemMAC;

AddRemMAC __self;

AddRemMAC* AddRemMAC_new(NetStream port1, NetStream port2);

// Method AddRemMAC.process1 (2) Derived from void process1(...)
void AddRemMAC_process1(__Packetp i) {
    __Packetp o = __Packet_concat2(__Packet_substring(i, 0, 14), i);
    NetStream_put(__self.out2, o);
}

// Method AddRemMAC.process2 (3) Derived from void process2(...)
void AddRemMAC_process2(__Packetp i) {
    __Packetp o = __Packet_subpacket(i, 14);
    NetStream_put(__self.out1, o);
}

// Constructor AddRemMAC
AddRemMAC* AddRemMAC_new(NetStream port1, NetStream port2) {
    int __i;
    for (__i = 0; __i < 17; __i++) {           Generating a method table
        __packetHandler[__i] = 0;
    }
    __packetHandler[port2] = &AddRemMAC_process2;
    __packetHandler[port1] = &AddRemMAC_process1;
    __self.out1 = port1;
    __self.out2 = port2;
    return &__self;
}           (4) Derived from the constructor
           (Public AddRemMAC(...))

// Main loop (Scheduler)
int __mainLoop(int no_ipd_wptr) {
    cvmx_wqe_t *wqe = NULL;

    // Omitted

    wait_for_link_up();

    // Omitted

    AddRemMAC_new(0, 16);           AddRemMAC object creation

    for (;;) {
        wqe = get_input_packet();
        if (wqe != NULL) {
            Repeating the following process
            for each packet (in wqe)

            // Omitted

            __Packetp __wqep;           Phonepl packet-pointer creation
            __wqep.u64 = 0;
            __wqep.s.pool = CVMX_FPA_WQE_POOL;
            __wqep.s.size = wqe->len;
            if (wqe->word2.s.bufs == 0) {
                /* if no buffered data (no data in DRAM) */
                __wqep.s.addr = cvmx_ptr_to_phys(wqe->packet_data);
                // *** IPv4/v6 cases? ***
            } else {
                Set_packet_representation(__wqep, CSP_CACHED);
                /* if data both in DRAM and in cache */
            }
            __wqep.s.addr = cvmx_ptr_to_phys(wqe);
            Set_packet_representation(__wqep, CSP_MIXED);           Tag insertion

            if (__packetHandler[wqe->ipprt]) {
                (*__packetHandler[wqe->ipprt])(__wqep);
            }

            Processing a packet by calling
            AddRemMAC_process1 or
            AddRemMAC_process2
        }
    }
    return 0;
}

```

Figure 3. Compiled code of MAC-header insertion/deletion program.

This program is explained instead of describing detailed compilation process because the process is too much complicated and the program structure can probably be used for other types of NPs. A compilation technique specialized for a singleton (*i.e.*, single-instance class) is applied to this program. Cores in an Octeon processor execute this program in parallel; that is, each core processes a packet. The program consists of five parts: part 1 derived from instance-variable declaration, parts 2 and 3 derived from methods process1 and process2, part 4 derived from the constructor, and part 5 derived from the main program.

In part 1, AddRemMAC type, which corresponds to instance of class AddRemMAC in Phonepl, is declared. Because a compiled object of class AddRemMAC has two objects of NetStream type, the corresponding structure components are declared. In parts 2 and 3, *i.e.*, method definitions, the element names in the source program are replaced by the element names in the run-time library. The run-time routines may be expanded in-line; however, they are not expanded in this example program. In part 4, *i.e.*, the constructor of class AddRemMAC, methods process1 and process2 are initialized. Assignment statements that correspond to the assignment statements in the source program are included in this part. In part 5, *i.e.*, the main program, the above constructor is called, and every time it receives a packet, one of the above two methods are called. Because NetStream type is an abstraction of a packet stream, the stream elements are handled one by one, and the scheduler for this process occupies the main part of part 5. When the function get_input_packet() is called, a packet is received, and the data representation of this packet is converted to that of +Net Phonepl by adding a tag, *i.e.*, cached (CSP_CACHED) or mixed (CSP_MIXED).

6. Evaluation

Both the programmability, especially ease of language use, and the performance of the implementation should be evaluated; however, because Phonepl is being improved, performance is focused in this evaluation. Two Phonepl programs for network-layer packet handling were written. Prototypes with these object programs were used for extending VNode, and the traffic was measured.

6.1. MAC-Header Addition/Deletion Program

The first program performs MAC-header addition/removal. It is a modified version of the program shown in **Figure 1**, and similar programs are used for extending virtualization-node (VNode) functions by using the node plug-in architecture [14]-[16]. Instead of duplicating the MAC header, the Phonepl program inserts a constant MAC header that contains fixed source and destination MAC addresses and a TEB type value (*i.e.*, transparent Ethernet bridge, x6558).

As shown in **Figure 4**, the above program was used in an extended VNode, which is a gateway between slices and external networks and is called NACE or NC [17]. This network consists of the VNode and two personal computers, PC1 and PC2. PC1 simulates a terminal or a virtual node in a slice. PC2 is in an external physical network. The VNode connects the slice and the external network, and it must convert the packet format, *i.e.*, convert from the internal to external protocols, and vice versa, but the base component of the VNode does not have this conversion function. The VNode is experimentally extended by the node plug-in architecture with the

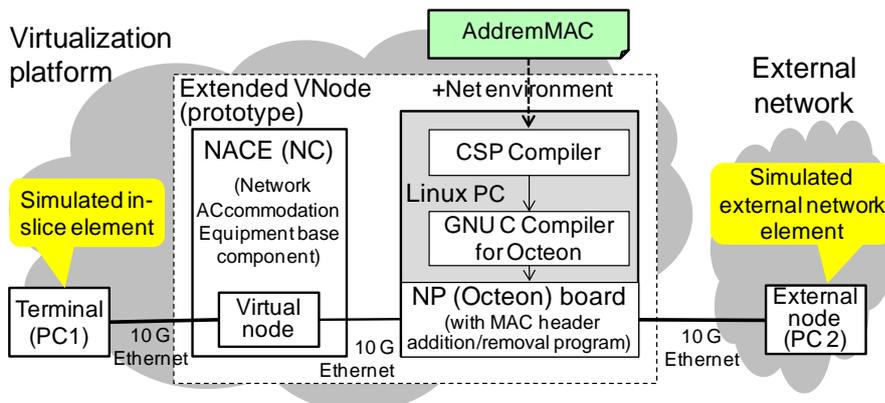


Figure 4. Extended VNode environment for experiments.

+Net environment, which consists of a PC with a Phonepl compiler, run-time routines, a GNU C compiler for Oocteon, and WANic-56512 with twelve-core 750-MHz Oocteon. By using conversion programs written in Phonepl, the VNode can adapt to various types of external networks.

Maximum performance of the test program was measured by using a network-measurement-tool suite called IXIA. Both operations, *i.e.*, MAC-header addition and deletion, were measured, and compared with a pass-through program, which is also written in Phonepl. The measurement results are shown in Figure 5. In this experiment, the input packet representation is mixed or cached, and the output packet representation is mixed or cached for header deletion and it is gathered for header addition. Uncached format is not used here because not whole cached data is removed by the header deletion. The maximum throughput (input rate) that can be passed with almost no packet drop is over 7.5 Gbps when the packet size is 256 bytes or larger. This throughput is close to the wire rate. The throughputs of two programs are mostly the same, indicating that the major overhead lies in the hardware or the initialization/finalization code, namely, not in the compiled code or the packet/string run-time routines.

Table 1 compares the performance of the Phonepl program on the Oocteon and a sequential C program on eight-core 3-GHz Intel Xeon processors. Although the performance of the former is much higher, it is mainly caused by the number of used cores. If all the cores are used, the throughput of Xeon may be better; however, it is very hard to use multiple cores and to preserve the order of packets in Xeon. As shown in Table 1, the Phonepl program is much shorter even when compared with the C program.

Moreover, Table 1 suggests an important difference between the two implementations; that is, the packet loss ratio is slowly increasing in the Xeon implementation because cache miss is slowly increasing, but packets are almost never lost if the input ratio is 9.2 Gbps or less in the Phonepl implementation because the memory usage is completely controlled.

6.2. Timestamp Handler for Network Virtualization Platform

The second program, which is described in detail in another paper [18], is a program for measuring communication delay between two points in the virtualization network. In this evaluation, NPs and the program was used only in VNodes, and a slow-path program was used in the gateways.

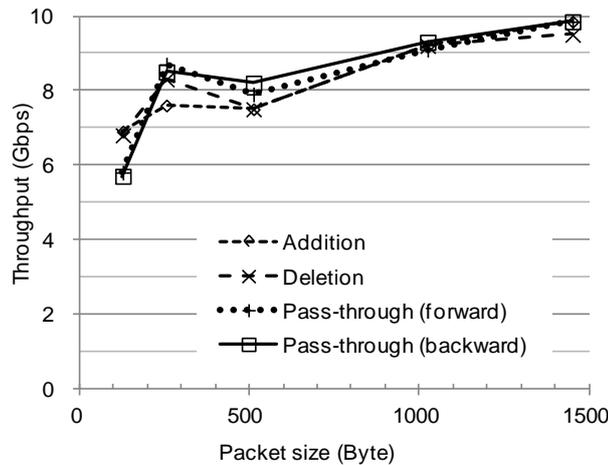


Figure 5. Performance of MAC-header addition/deletion.

Table 1. Results of MAC header addition/deletion.

| Implementation | Throughput (Gbps)* | | Program lines |
|---------------------------------|---------------------------------------|---------------------------------------|------------------|
| | Header addition | Header deletion | |
| Phonepl program | 9.2 [†] | 9.2 [†] | 26 [‡] |
| C program (Xeon, single core)** | 2.3 [†] (4.0 ^{††}) | 1.7 [†] (4.0 ^{††}) | 161 [‡] |

*Packet size: 1024 B; **Promiscuous mode is used; [†]No packet loss (ratio < 10⁻⁶); ^{††}Packet loss ratio = 10⁻³; [‡]Comment-only lines are not counted.

A VNode platform can support delay measurement function without adding programs and data (*i.e.*, packet format) for measurement to programs in virtual nodes. This function is useful when slice developers want to measure delay of a high-bandwidth application with certain intelligent functions in relaying nodes. A special type of virtual links between nodes, which is called measurable VLAN virtual link (MVL) type and developed by using the VNode plug-in architecture, is used to implement this function. MVLs are implemented by using timestamp insertion/deletion programs in the nodes. A VNode removes the platform header, which includes a GRE/IP or VLAN header and the timestamp, from an incoming packet and adds one to an outgoing packet, so programs that handles packets on a slice never see the platform header.

The virtualization-network structure used for this experiment is drawn in **Figure 6**. Two terminals communicate using a slice. The physical network contains two VNodes. Each VNode contains a virtual node, which are connected by an MVL. In the platform, each packet has a platform header with a timestamp.

The communication and measurement methods used for this experiment is as follows. The timestamp is inserted at the entrance gateway. Each VNode generates a packet for the virtual node by removing the platform header from an incoming packet and restores the timestamp to outgoing packets that comes from the virtual node and are identified with a stored incoming packet. The timestamp is tested and deleted at the exit gateway, which calculates the delay between the entrance and exit gateways. In the network described in **Figure 6**, the two VNodes and one PC is used for the two gateways (and terminals) to avoid the difficult synchronization problem. Terminal PCs communicate each other by using Ethernet packets, which are switched by the MAC addresses in the virtual nodes. A WANic-56512 that contains the program handles both incoming and outgoing packets. An Ethernet switch program, which is a slow-path program, works on a virtual node in a VNode.

The NP also swaps the external and internal MAC addresses in the platform header [1]. To swap addresses, the program contains a conversion table for these MAC addresses, which is implemented using a string array, and accepts virtual-link-creation and deletion requests. A creation request adds an entry to the conversion table.

The results show the gateway-to-gateway delay is 178 μ S ($\sigma = 24 \mu$ S). **Table 2** compares the performance of the 750 MHz Octeon and the 3-GHz Xeon. The performance is very close to wire rate. The C program is relatively short because this program does not contain conversion-table configuration code but the Phonepl program contains it. However, the former is still much longer.

7. Concluding Remarks

An open, portal, and high-level language, called Phonepl, is proposed. By using Phonepl, a programmer can develop a program that uses SRAM and DRAM appropriately without having to be aware of a distinction between SRAM and DRAM. To handle packets appropriately in this environment, four packet data-representations and packet-operation methods are proposed. A prototype using Octeon NP was developed and evaluated. The

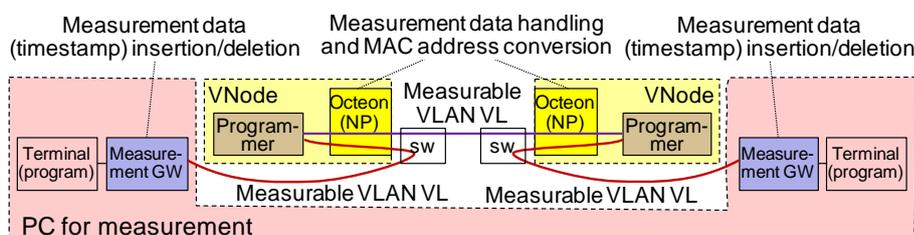


Figure 6. Virtualization-network structure for time-stamp handling.

Table 2. Results of timestamp handling and conversion.

| Implementation | Throughput (Gbps) [*] | | Program lines |
|---|---------------------------------------|---------------------------------------|------------------|
| | Header addition | Header deletion | |
| Phonepl program | 10.0 [†] | 9.5 [†] | 99 [‡] |
| C program (Xeon, single core) ^{**} | 2.3 [†] (4.0 ^{††}) | 2.2 [†] (4.0 ^{††}) | 190 [‡] |

^{*}Packet size: 1024 B; ^{**}Promiscuous mode is used; [†]No packet loss (ratio < 10⁻⁶); ^{††}Packet loss ratio = 10⁻³; [‡]Comment-only lines are not counted.

throughput of the prototype system is close to the wire rate, *i.e.*, 10 Gbps, when the packet size is 256 bytes or larger, in several packet-conversion applications. Although this is a preliminary result, it proves the proposed method is promising in achieving our objectives, *i.e.*, popularity among developers, reduced cost in programmability, and portability.

Future work includes evaluation of Phonepl language and processor by human programmers and improvement of the language design and implementation according to the evaluation result. Although Phonepl and the language processor inevitably have limitations, they should be acceptable and, if possible, natural to programmers. Future work also includes implementation of Phonepl for other types of NPs to prove the portability. Moreover, the memory allocation and deallocation mechanism must be improved to reduce memory leak caused by global variable assignments and the performance of the Phonepl language processor should be improved.

Acknowledgements

The author thanks Professor Aki Nakao from the University of Tokyo, Yasushi Kasugai, Kei Shiraishi, Takanori Ariyoshi, Takeshi Ishikura, and Toshiaki Tarui from Hitachi, Ltd., and other members of the project for discussions and evaluations. Part of the research results described in this paper is an outcome of the Advanced Network Virtualization Platform Project A funded by National Institute of Information and Communications Technology (NICT).

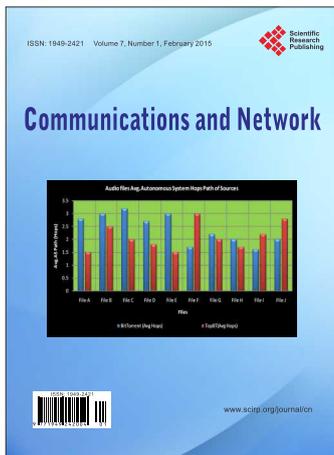
References

- [1] Kanada, Y., Shiraishi, K. and Nakao, A. (2012) Network-Virtualization Nodes That Support Mutually Independent Development and Evolution of Components. *IEEE International Conference on Communication Systems (ICCS 2012)*. <http://dx.doi.org/10.1109/iccs.2012.6406171>
- [2] Shah, N., Plishker, W., Ravindran, K. and Keutzer, K. (2004) NP-Click: A Productive Software Development Approach for Network Processors. *IEEE Micro*, **24**, 45-54. <http://dx.doi.org/10.1109/mm.2004.53>
- [3] Fatahalian, K., Knight, T.J., Houston, M., Erez, M., Horn, D.R., Leem, L., Park, J.Y., Ren, M., Aiken, A., Dally, W. J. and Hanrahan, P. (2006) Sequoia: Programming the Memory Hierarchy. *2006 ACM/IEEE Conference on Supercomputing*. <http://dx.doi.org/10.1109/sc.2006.55>
- [4] Goglin, S.D., Hooper, D., Kumar, A. and Yavatkar, R. (2003) Advanced Software Framework, Tools, and Languages for the IXP Family. *Intel Technology Journal*, **7**, 64-76.
- [5] Cavium Networks (2010) OCTEON Programmer's Guide. The Fundamentals. http://university.caviumnetworks.com/downloads/Mini_version_of_Prog_Guide_EDU_July_2010.pdf
- [6] Bell, S., Edwards, B., Amann, J., Conlin, R., Joyce, K., Leung, V., MacKay, J., Reif, M., Bao, L.W., Brown, J., Mattina, M., Miao, C.-C., Ramey, C., Wentzlaff, D., Anderson, W., Berger, E., Fairbanks, N., Khan, D., Montenegro, F., Stickney, J. and Zook, J. (2008) TILE64-Processor: A 64-Core SoC with Mesh Interconnect. *IEEE International Solid-State Circuits Conference (ISSCC 2008)*, San Francisco, 3-7 February 2008, 88-598. <http://dx.doi.org/10.1109/isscc.2008.4523070>
- [7] Chen, M.K., Li, X.F., Lian, R., Lin, J.H., Liu, L.X., Liu, T. and Ju, R. (2005) Shangri-La: Achieving High Performance from Compiled Network Applications While Enabling Ease of Programming. *2005 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'05)*, 224-236. <http://dx.doi.org/10.1145/1064978.1065038>
- [8] Kohler, E., Morris, R., Chen, B.J., Jannotti, J. and Frans Kaashoek, M. (2000) The Click Modular Router. *ACM Transactions on Computer Systems*, **18**, 263-297. <http://dx.doi.org/10.1145/354871.354874>
- [9] Foster, N., Harrison, R., Freedman, M.J., Monsanto, C., Rexford, J., Story, A. and Walker, D. (2011) Frenetic: A Network Programming Language. *16th ACM SIGPLAN International Conference on Functional Programming (ICFP'11)*. <http://dx.doi.org/10.1145/2034773.2034812>
- [10] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S. and Turner, J. (2008) Open Flow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, **38**, 69-74. <http://dx.doi.org/10.1145/1355734.1355746>
- [11] Arasu, A., Babu, S. and Widom, J. (2006) The CQL Continuous Query Language: Semantic Foundations and Query Execution. *The VLDB Journal*, **15**, 121-142. <http://dx.doi.org/10.1007/s00778-004-0147-z>
- [12] Monsanto, C., Foster, N., Harrison, R. and Walker, D. (2012) A Compiler and Run-Time System for Network Programming Languages. *Proceedings of the 39th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, Philadelphia, 25-27 January 2012, 217-230. <http://dx.doi.org/10.1145/2103656.2103685>
- [13] Nakao, A. (2012) VNode: A Deeply Programmable Network Testbed through Network Virtualization. *Proceedings of*

the 3rd IEICE Technical Committee on Network Virtualization, Tokyo, 2 March 2012.

<http://www.ieice.org/~nv/05-nv20120302-nakao.pdf>

- [14] Kanada, Y. (2013) A Node Plug-In Architecture for Evolving Network Virtualization Nodes. 2013 *Software Defined Networks for Future Networks and Services (SDN4FNS)*. <http://dx.doi.org/10.1109/sdn4fns.2013.6702531>
- [15] Kanada, Y. (2014) A Method for Evolving Networks by Introducing New Virtual Node/Link Types Using Node Plug-Ins. *Proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, Krakow, 5-9 May 2014, 1-8. <http://dx.doi.org/10.1109/noms.2014.6838417>
- [16] Kanada, Y. (2014) Controlling Network Processors by Using Packet-Processing Cores. *Proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Victoria, 13-16 May 2014, 690-695. <http://dx.doi.org/10.1109/waina.2014.112>
- [17] Kanada, Y., Shiraishi, K. and Nakao, A. (2012) High-Performance Network Accommodation into Slices and In-Slice Switching Using a Type of Virtualization Node. *2nd International Conference on Advanced Communications and Computation (Infocomp 2012)*, IARIA.
- [18] Kanada, Y. (2014) Extending Network-Virtualization Platforms Using a Specialized Packet-Header and Node Plug-Ins. *Proceedings of the 22nd International Conference on Telecommunications and Computer Networks*, Split, 17-19 September 2014.



Communications and Network

ISSN 1949-2421 (print) ISSN 1947-3826 (online)

<http://www.scirp.org/journal/cn>

CN, an international journal, dedicates to the latest advancement of communications and network technologies. The goal of this journal is to keep a record of the state-of-the-art research and promote the research work in these fast moving areas.

All manuscripts submitted to CN must be previously unpublished and may not be considered for publication elsewhere at any time during CN's review period. Additionally, accepted ones will immediately appear online followed by printed in hard copy. The topics to be covered by Communications and Network include, but are not limited to:

- ◆ Algorithms and Programs
- ◆ Applications and Value-Added Services
- ◆ Cognitive Radio
- ◆ Communication Protocols
- ◆ Complexity
- ◆ Computer and Network Security
- ◆ Cooperative Communications
- ◆ Data and Information Security
- ◆ Error Correcting Codes
- ◆ Fading/Equalization
- ◆ Green Technology
- ◆ Hardware and Software System Architecture
- ◆ High Performance Computing, Grid Computing, Cloud Computing
- ◆ Internet/Intranet Protocols
- ◆ Internet/Intranet Services
- ◆ Location Based Services
- ◆ Mobile and Portable Communications Systems
- ◆ Mobile Computing Systems
- ◆ Mobility Networks and Protocols
- ◆ Modulation/Signal Design
- ◆ Multi Antenna/User Systems
- ◆ Multimedia Services in Wireless Networks
- ◆ Network Protocol, QoS and Congestion Control
- ◆ Operations and Management
- ◆ Optical Communications
- ◆ Overlay and Peer-to-Peer Networks
- ◆ Self-Stabilization, Autonomic Computing
- ◆ Shared and Transactional Memory, Synchronization Protocols, Concurrent Programming
- ◆ Simulation/Analytical Evaluation of Communication Systems
- ◆ Source Coding/Data Compression
- ◆ Switching and Routing
- ◆ Synchronization
- ◆ Traffic Diagnosis
- ◆ Transmission and Access Systems
- ◆ Wired and Wireless Integration
- ◆ Wireless Network Design and Performance Evaluation

We are also interested in short papers (letters) that clearly address a specific problem, and short survey or position papers that sketch the results or problems on a specific topic. Authors of selected short papers would be invited to write a regular paper on the same topic for future issues of the CN.

Website and E-Mail

<http://www.scirp.org/journal/cn>

E-Mail: cn@scirp.org

What is SCIRP?

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

What is Open Access?

All original research papers published by SCIRP are made freely and permanently accessible online immediately upon publication. To be able to provide open access journals, SCIRP defrays operation costs from authors and subscription charges only for its printed version. Open access publishing allows an immediate, worldwide, barrier-free, open access to the full text of research papers, which is in the best interests of the scientific community.

- High visibility for maximum global exposure with open access publishing model
- Rigorous peer review of research papers
- Prompt faster publication with less cost
- Guaranteed targeted, multidisciplinary audience



**Scientific
Research
Publishing**

Website: <http://www.scirp.org>

Subscription: sub@scirp.org

Advertisement: service@scirp.org