

Transactions on Networks and Communications

ISSN: 2054-7420

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD	I
DISCLAIMER	II
Exploring Network Softwarization and Virtualization by Applying SDN/NFV to 5G and IoT Bao-Shuh Paul Lin, Yi-Bing Lin, Li-Ping Tung, Fuchun Joseph Lin	1
Optimizing TCP Goodput and Delay in Next Generation IEEE 802.11 (ax) Devices Oran Sharon	14
Multiplexing and De Multiplexing using Visual Basic Program Ahmad A. Mohammed AL-Mafrji	41

Editor In Chief

Dr Patrick J Davies
Ulster University, United Kingdom

EDITORIAL ADVISORY BOARD

Professor Simon X. Yang
The University of Guelph
Canada

Professor Shahram Latifi
Dept. of Electrical & Computer Engineering University of
Nevada, Las Vegas
United States

Professor Farouk Yalaoui
University of Technology of Troyes
France

Professor Julia Johnson
Laurentian University, Sudbury, Ontario
Canada

Professor Hong Zhou
Naval Postgraduate School Monterey, California
United States

Professor Boris Verkhovskiy
New Jersey Institute of Technology, New Jersey
United States

Professor Jai N Singh
Barry University, Miami Shores, Florida
United States

Professor Don Liu
Louisiana Tech University, Ruston
United States

Dr Steve S. H. Ling
University of Technology, Sydney
Australia

Dr Yuriy Polyakov
New Jersey Institute of Technology, Newark
United States

Dr Lei Cao
Department of Electrical Engineering, University of
Mississippi
United States

Dr Kalina Bontcheva
Dept. of Computer Science
University of Sheffield, *United Kingdom*

Dr Bruce J. MacLennan
University of Tennessee, Knoxville, Tennessee
United States

Dr Panayiotis G. Georgiou
USC university of Southern California, Los Angeles
United States

Dr Armando Bennet Barreto
Dept. Of Electrical and Computer Engineering
Florida International University
United States

Dr Christine Lisetti
School of Computing and Information Sciences
Florida International University
United States

Dr Youlian Pan
Information and Communications Technologies
National Research Council *Canada*

Dr Xuewen Lu
Dept. of Mathematics and Statistics
University of Calgary
Canada

Dr Sabine Coquillart
Laboratory of Informatics of Grenoble
France

Dr Claude Godart
University of Lorraine
France

Dr Paul Lukowicz
German Research Centre for Artificial Intelligence
Germany

Dr Andriani Daskalaki
Max Planck Institute for Molecular Genetics
MOLGEN
Germany

Dr Jianyi Lin
Department of Computer Science
University of Milan, *Italy*

Dr Hiroyuki SATO
Information Technology Centre
The University of Tokyo
Japan

Dr Christian Cachin
IBM Research – Zurich
Switzerland

Dr W. D. Patterson
School of Computing, Ulster University
United Kingdom

Dr Alia I. Abdelmoty
Cardiff University, Wales
United Kingdom

Dr Sebastien Lahaie
Market Algorithms Group, Google
United States

Dr Jenn Wortman Vaughan
Microsoft
United States

Dr Jianfeng Gao
Microsoft
United States

Dr Silviu-Petru Cucerzan
Machine Learning Department, Microsoft
United States

Dr Ofer Dekel
Machine Learning and Optimization Group, Microsoft
Israel

Dr K. Ty Bae
Department of Radiology
University of Pittsburgh
United States

Dr Jiang Hsieh
Illinois Institute of Technology
University of Wisconsin-Madison
United States

Dr David Bulger
Department of Statistics
MACQUARIE University
Australia

Dr YanXia Lin
School of Mathematics and Applied Statistics
University of Wollongong
Australia

Dr Marek Reformat
Department of Electrical and Computer Engineering
University of Alberta
Canada

Dr Wilson Wang
Department of Mechanical Engineering
Lake head University
Canada

Dr Joel Ratsaby
Department of Electrical Engineering and Electronics
Ariel University
Israel

Dr Naoyuki Kubota
Department of Mechanical EngineeringTokyo
Metropolitan University
Japan

Dr Kazuo Iwama
Department of Electrical Engineering
Koyoto University
Japan

Dr Stefanka Chukova
School of Mathematics and Statistics
Victoria University of Wellington
New Zealand

Dr Ning Xiong
Department of Intelligent Future Technologies
Malardalen University
Sweden

Dr Khosrow Moshirvaziri
Department of Information systems
California State University Long Beach
United States

Dr Kechen Zhang
Department of Biomedical Engineering
Johns Hopkins University
United States

Dr. Jun Xu
Sun Yat-Sen University , Guangzhou
China

Dr Dinie Florancio
Multimedia Interaction and Collaboration Group
Microsoft
United States

Dr Jay Stokes
Department of Security and Privacy, Microsoft
United States

Dr Tom Burr
Computer, Computational, and Statistical Sciences Division
Los Alamos National Laboratory
United States

Dr Philip S. Yu
Department of Computer Science
University of Illinois at Chicago
United States

Dr David B. Leake
Department of Computer Science
Indiana University
United States

Dr Hengda Cheng
Department of Computer Science
Utah State University
United States

Dr. Steve Sai Ho Ling
Department of Biomedical Engineering
University of Technolia Sydney
Australia

Dr. Igor I. Baskin
Lomonosov Moscow State University,
Moscow
Russian Federation

Dr. Konstantinos Blekas
Department of Computer Science & Engineering,
University of Ioannina
Greece

Dr. Valentina Dagiene
Vilnius University
Lithuania

Dr. Francisco Javier Falcone Lanás
Department of Electrical Engineering,
Universidad Publica de Navarra, UPNA
Spain

Dr. Feng Lin
School of Computer Engineering
Nanyang Technological University
Singapore

Dr. Remo Pareschi
Department of Bioscience and Territory
University of Molise
Italy

Dr. Hans-Jörg Schulz
Department of Computer Science
University of Rostock
Germany

Dr. Alexandre Varnek
University of Strasbourg
France

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

Exploring Network Softwarization and Virtualization by Applying SDN/NFV to 5G and IoT

Bao-Shuh Paul Lin^{1,2}, Yi-Bing Lin², Li-Ping Tung¹, Fuchun Joseph Lin²

¹*Microelectronics & Information Research Center, National Chiao Tung University, Hsinchu, Taiwan*

²*Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan*

bplin@nctu.edu.tw, liny@nctu.edu.tw, lptung@nctu.edu.tw, fjlin@nctu.edu.tw

ABSTRACT

The development trends of networking and wireless communications technologies have evolved toward Open Networking (ON), Software Defined Networks (SDN), Network Function Virtualization (NFV), and Disaggregation. ON and SDN/NFV are playing the key roles for softwarization and virtualization (S&V) of next generation networks such as 5th Generation Communications and Networks (5G) and Internet of Things (IoT). This article explores the S&V of 5G & IoT and addresses the challenging issues related to applying SDN/NFV to 5G and IoT. These issues include system performance, end-to-end interoperability, software compatibility, global deployment, integration of SDN and NFV, and shortage of talents. In addition, we describe the research and development status of adapting ON architecture and applying SDN/NFV to 5G and IoT in Taiwan. Finally we report the collaboration between National Chiao Tung University (NCTU) and networking industry in the areas of programmable switches built on top of silicon with Programming Protocol-independent Packet Processor (P4) high-level script language and prototyping an OpenAirInterface (OAI)-based M-CORD.

Keywords: 5th Generation Communications and Networks (5G), Internet of Things (IoT), Software Defined Networks (SDN), Network Function Virtualization (NFV), Programming Protocol-independent Packet Processor (P4), Central Office Re-architected Data center (CORD), Softwarization & Virtualization (S&V), OpenAirInterface (OAI)

1 Introduction.

There has been a paradigm shift in the networking industry, from a proprietary networking with vendor-dependent software/hardware to vendor-independent open source and white-box switches, as illustrated in Figure 1. The white box is the commodity hardware which is available from network equipment ODM vendors. On the other hand, the open source software executed on the commodity hardware in the control plane can be very complex, which consist of controller (with various OSs, ONOS, ODL, RYU) [17][19], Network Function Virtualization (NFV), and Cloud OS. Not surprisingly, the software engineers involved in the realization of this paradigm shift often encounter various challenges, issues, or difficulties.

Besides the control plane of Software Defined Networking (SDN) as depicted in Figure 2(a), the software of data plane is also under rapid development. In particular, the new invention of switch programming language called Programming Protocol-independent Packet Processor (P4) that can be tightly coupled

DOI: 10.14738/tnc.64.4825

Publication Date: 17th July 2018

URL: <http://dx.doi.org/10.14738/tnc.64.4825>

with SDN controller, Open network (ON) architecture, and Tofino [16][23] system on chip (SoC) has brought lots of attention to the open networking community. Figure 2(b) shows the relationships among P4, Central Office Re-architected a Data Center (CORD), and Open Network Operating System (ONOS) based on a layered architecture. Among them P4 is a high-level programming language designed to configure/re-configure hardware, CORD is the architecture to transform telecommunications central offices to becoming data centers and ONOS [20] is the open networking OS developed by ONF. The ecosystem of this ON community includes network service operators, network equipment vendors, silicon providers, and white box ODM providers become so active. In the meantime, the next generation 5G communications system will be available within 2-3 years while the applications of IoT will appear in various industries.

The impacts of new ON architecture includes high speed network vendor (P4 switch), Rack solution provider (M-CORD), silicon provider (SoC/IC), commodity provider (white-box switches, servers, I/O blades), network service operators (services) are becoming quite significant. P4 switch provides high speed switch with highest programmability which capability is embedded in SoC/IC. CORD architecture will convert the central offices of telecommunications network to data center which can provide fast and rich services. Consequently, this conversion can reduce the capital expense (CAPEX) and operations expense (OPEX) for network service operator.

With the efforts of research and development related to SDN/NFV, open source for open networking, we have encountered and identified six challenging issues: performance, interoperability, compatibility, deployment, integration of SDN and NFV [4][5], and shortage of talents. In this paper, we will also address how to tackle these challenges.

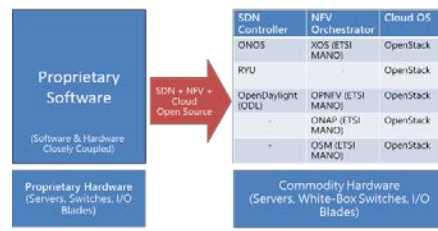


Figure 1 Open Networking Trend

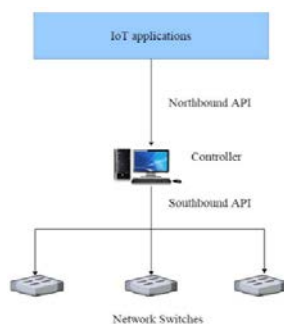


Figure 2(a) SDN Architecture

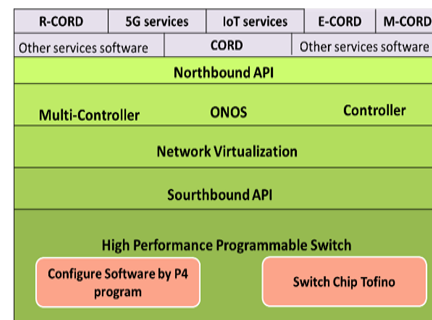


Figure 2(b) SDN Layered Architecture based on P4/ONOS/CORD

Figure 2 SDN layered architecture based on P4/CORD/ONOS

2 Network Softwarization & Virtualization

There are several papers [1][2][7][8] pay much attention to realize SDN for IoT and virtualization of network functions of network elements of IoT. The other papers [6][9][10][11] are investigating how to take care of 5G or mobile broadband. In this paper, we address both of IoT and 5G from softwarization and virtualization point of view.

SDN sets the standards for converting a network based on propriety hardware/software into a software defined network based on commodity hardware and open source. SDN defines a network with the separation of control plane (controller) and a data plane (switch). Figure 2 shows the SDN layered Architecture where the control of network switches is separated out and moved to SDN controllers such as ONOS via defining southbound API such as OpenFlow. Also, the P4 programming language is defined to enable the data plane programmability of SDN switches. Via P4 an SDN controller can configure and re-configure switches during runtime.

On the other hand, NFV provide the ways to virtualize the network elements by software components or modules. According to NetSoft 2018 conference statement [26] and ONF (Open Networking Foundation), SDN, NFV, and Cloud Edge computing have been driving a paradigm shift in the telecommunications and ICT industries. Network softwarization and programmability promise to reduce OPEX and to provide better flexibility and bring new services to their customers sooner [21].

3 P4 and CORD Becoming Popular and Their Impacts

Since ONOS is available, the P4 and CORD are becoming popular in the ON community. This creates significant impacts to the SDN ecosystem including high speed network vendor (P4 switch), Rack solution provider (M-CORD box) silicon provider (SoC/IC), commodity provider (white box switch, server, storage I/O, blade), and network service operator (services)

3.1 P4 as an SDN Switch Programming Language

P4 [12][13][14][15] is a re-configurable, multi-platform, protocol and target independent, packet processing language which is used to facilitate dynamically programmable, extensible, packet processing in the SDN data plane. In SDN, a switch uses a set of “match+action” flow tables to apply rules for packet processing and P4 provides an efficient way to configure the packet processing pipelines to achieve this goal. Specifically a packet consists of a packet header and payload, and the header includes several fields defined by the network protocol. A P4 program describes how packet headers are parsed and their fields are processed by using the flow tables where matched operations may modify the header fields of packet or the content of metadata registers.

Figure 3 shows how P4 make SDN switch more flexible and programmable [12][13]. Figure 3(a) shows SDN switch without supporting P4 while Figure 3(b) with P4 support.

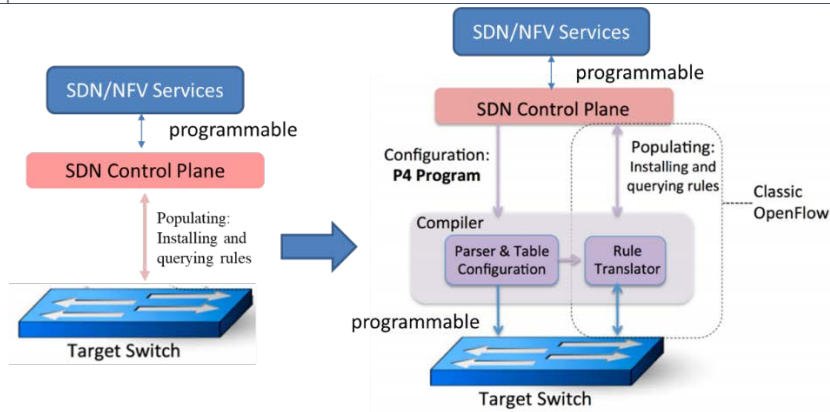


Figure 3(a) Switch without P4 support Figure 3(b) Switch with P4 support

Figure 3 SDN Switch with/without P4 support

We have conducted the P4 switch system performance evaluation by developing two applications :

- (1) Traffic Classification for Dynamic QoS Control based on P4 Switch[24]
- (2) Dynamic Load Balancing and Congestion Avoidance based on P4 Switch [25]

Figure 4(a), 4(b), and 4(c) show the topology of experiment, architecture, and the sample percentage of applications distribution respectively for the traffic classification for dynamic QoS control based on P4 switch, Figure 4(d) and 4(e) illustrates the two topologies and the architecture respectively for dynamic load balancing and congestion avoidance based on P4 switch.

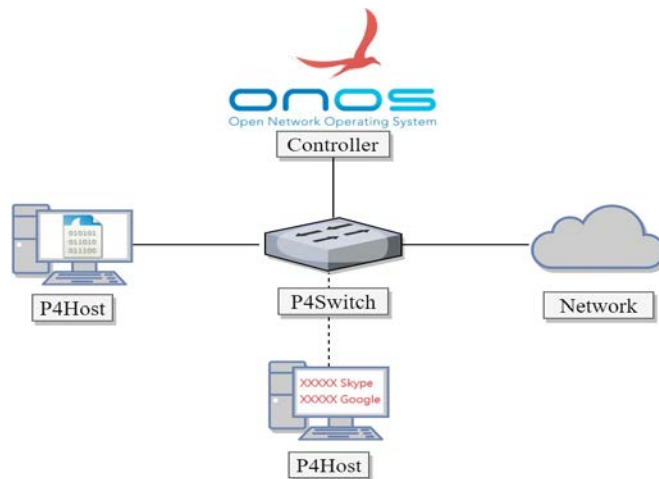


Figure 4(a) Topology of Experiment for Traffic Classification for Dynamic QoS Control

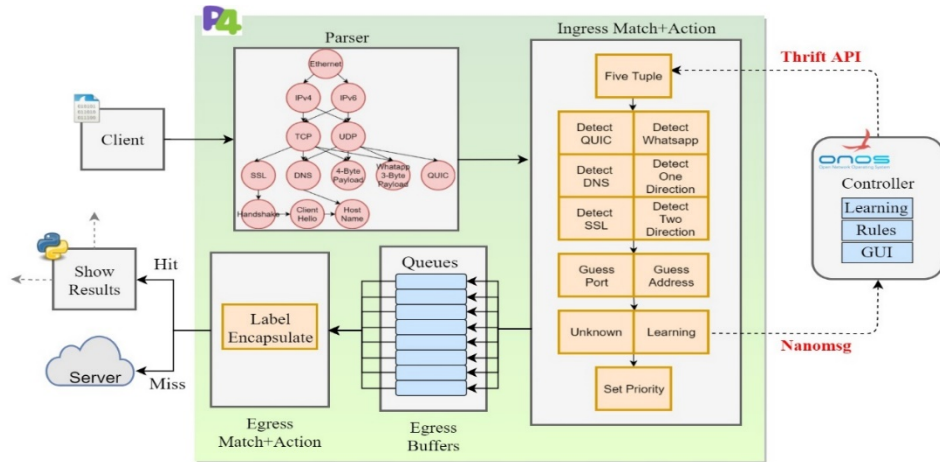


Figure 4(b) Architecture of P4-based Traffic Classification for Dynamic QoS Control

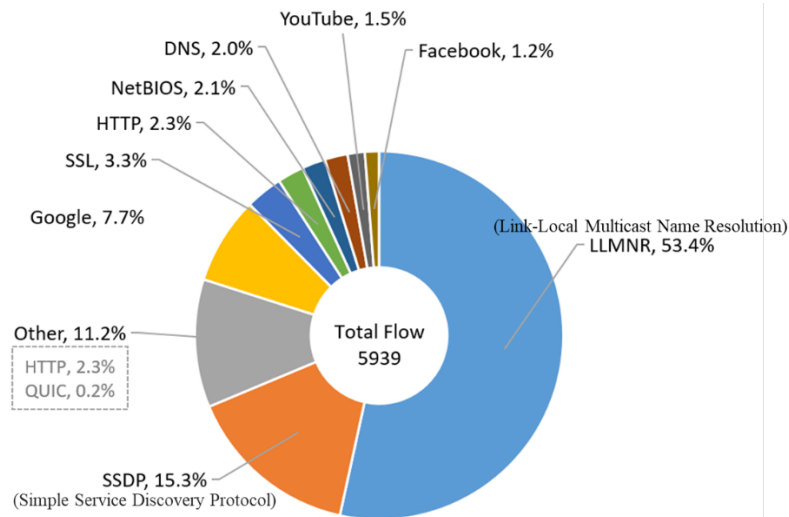


Figure 4(c) Percentage of applications for a sample traffic flow

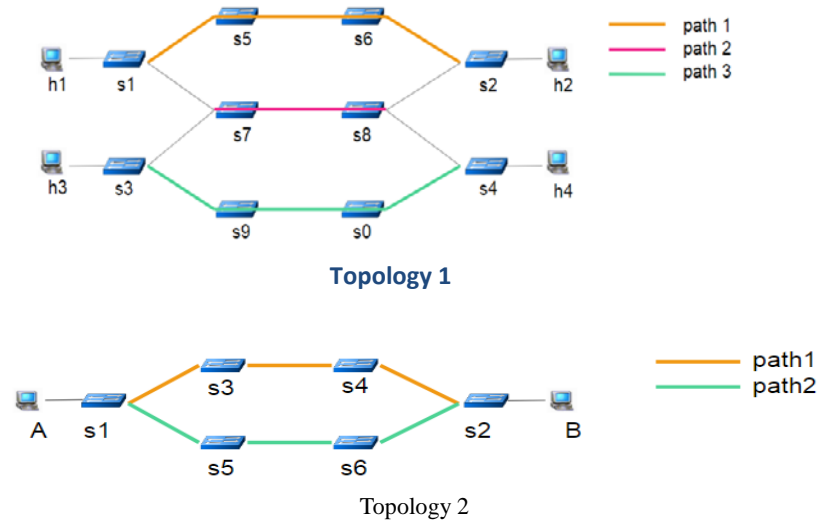


Figure 4(d) Topology1&2 for Dynamic Load Balancing and Congestion Avoidance

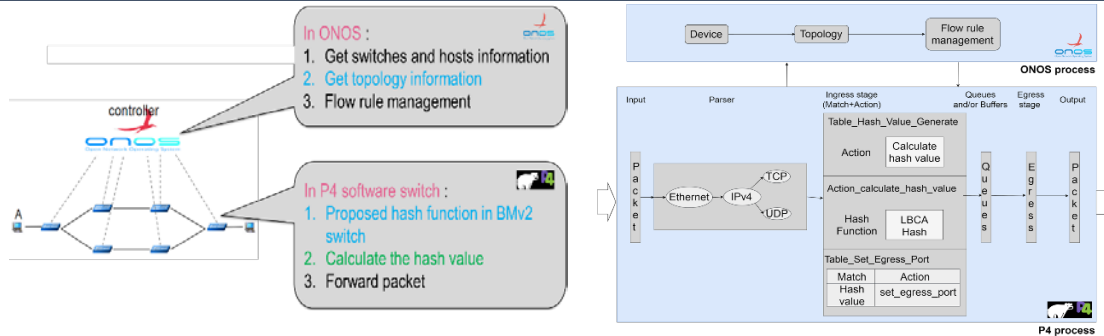


Figure 4(e) Architecture for Dynamic Load Balancing and Congestion Avoidance

CORD = SDN x NFV x Cloud



Figure 5 Basic CORD Architecture [18] [20]

3.2 CORD as CO Re-architected Data Center

The basic CORD architecture [18] [20] transforms telecommunications Central Office (CO) to Data Center (DC) as illustrated in Figure 5. This transformation will change the CO with propriety hardware/software to DC with white-box commodity and open source. Based on this CORD architecture, many possible applications may apply SDN/NFV and Cloud (local cloud and remote cloud) to telecommunications networks, including:

- a) Mobile/Wireless network – M-CORD
- b) Residential/wireline network – R-CORD
- c) Enterprise network – E-CORD

4 Why 5G and IoT ?

5G is the next generation communications networks which provides very high data rate (throughput) with very low latency. SDN, NFV, and mm-wave are claimed as 3 key ingredients for 5G. In this article, we are trying to find out that 5G is capable of handling various requirements of IoT applications and services. The IoT applications may generate big data or small data and services may be mission critical. 5G network architecture is designed to handle these IoT applications and services properly.

The international standard organization, Next Generation Management Network (NGMN) has proposed three real and existing categories of network slicing for 5G including smart phones, autonomous driving, and massive IoT as shown in Figure 6. To zoom into massive IoT category, it can be further divided into as many network slicing as shown in Figure 7. Figure 8 indicates the realization of 5G network slicing [22] by SDN/NFV technology through the virtualization of RAN and EPC

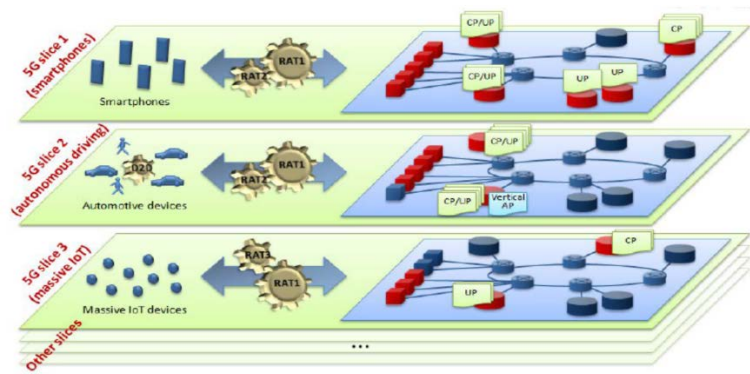


Figure 6 NGMN proposes three categories of network slicing

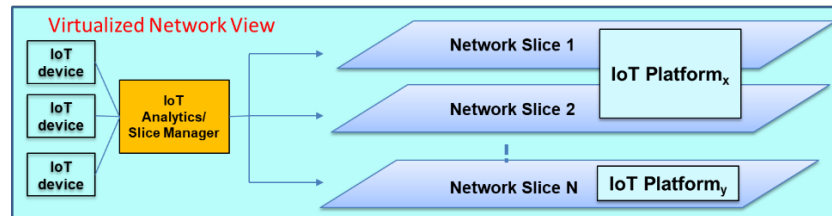


Figure 7 Applying SDN/NFV to IoT Services to realize dynamic deployment of IoT services

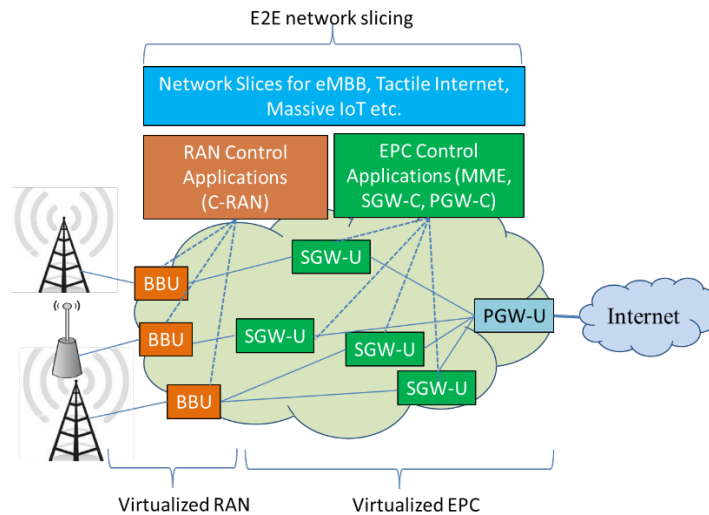


Figure 8 Realizing 5G network slicing by SDN/NFV

5 Challenges and Technical Issues

Through the joint Research and Development of SDN/NFV technology between CHT-TL and NCTU, we have discovered and identified six technical and human resource issues [3] [4] [8]. In the section, we also address how to resolve those issues.

5.1 Performance for Network based on ON/Open Source vs. Propriety

The network performance comparison is an issue between approach with ON & open source verse approach with propriety hardware & software. But if the network scalability and programmability are the requirements, then ON/open source approach or solution is a better choice. Also, the network performance of ON/open source approach continuous to improve automatically while the propriety approach needs to pay more to buy a new model with better capacity and capability.

5.2 End-to-End Interoperability

The choice of SDN popular controller's OSs includes ONOS, Open DayLight (ODL), RYU. The available switches from various vendors built on top of different SoC/IC solutions. How to test the end-to-end interoperability among SDN of north Taiwan Universities and to connect to global are shown in Figure 9. It indicates the network topology, controller OS of each site and various brands of switches of each site:

- a) NCTU SDN controller runs ONOS as its OS with switches by PICA 8, Edgecore, and Broadcom,
- b) CHT-TL (CHT: Taiwan top one telecom operator) controller runs RYU as its OS and with Switches by NEC
- c) NTHU's controller runs ODL as its OS with switches by PICA 8
- d) Kreonet and AmLight are global links to South Korea and to Florida then to South America

Through those intensive exercises and experiments following the successful global deployment led by the lab team of ONF, the issue of interoperability is resolved.

5.3 Global Deployment

To deploy SDN global peering, NCTU team worked together with ONF Lab Team (used to call ON.LAB). There are 8 cities around 5 Continents participated in this global SDN-IP Peering. The result was demo at Open Network Summit (ONS) went very well. Figure 10 and Figure 11 provide two screens shown during the demo.

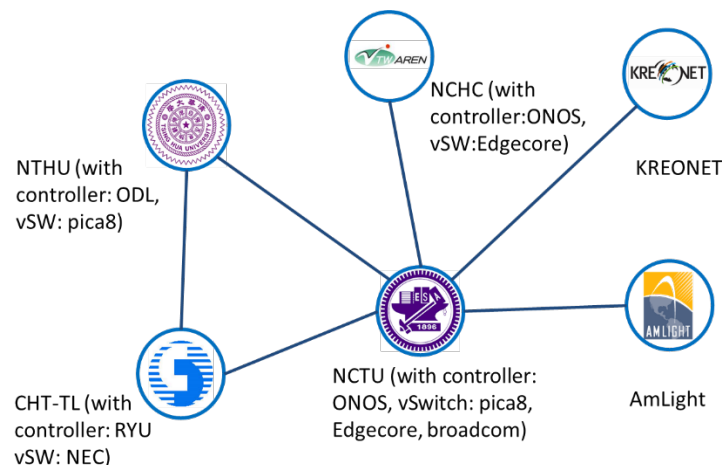


Figure 9 Network Topology in North Taiwan (Link to Global)

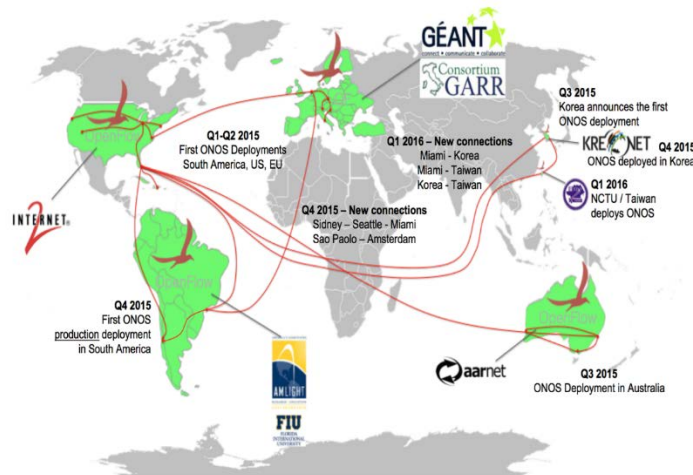


Figure 10 SDN-IP Global Peering Deployment with 5 Continents

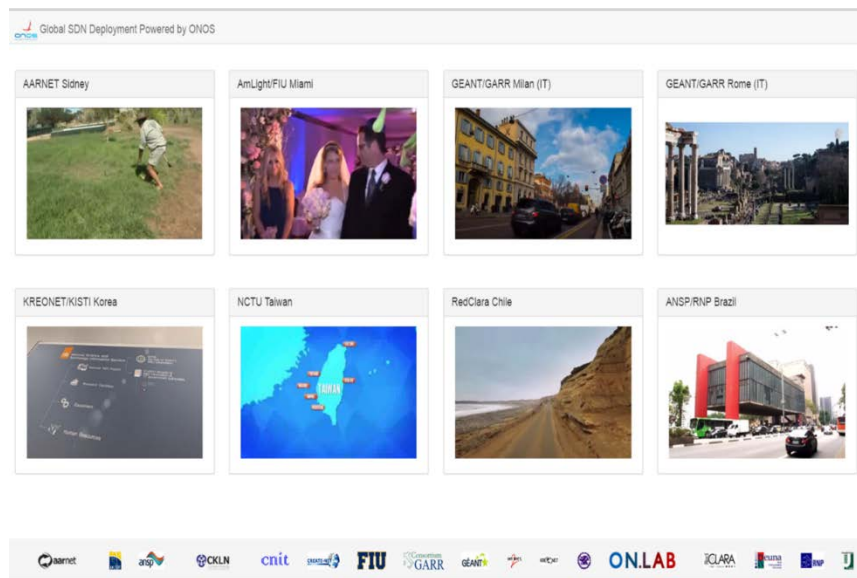


Figure 11 Video Shown from 8 Cities and Regions during SDN-IP Global Deployment

5.4 Software Compatibility

To get familiar with the availability and functionality of the proper version released open source software and tools in particular related to open networking software is essential to resolve the incompatible issue. The version control and development methodology are also very important.

5.5 Vertical Integration

The challenge of integration of SDN and NFV was addressed by several papers [3][4]. By creating vertically-integrated stack as SDN distribution. The software and tools of open source related to SDN will continue to update or revise and the virtualization of network elements defined in NFV may change or update time to time. The case In Figure 12 shows a successful vertically integrated as SDN distribution.

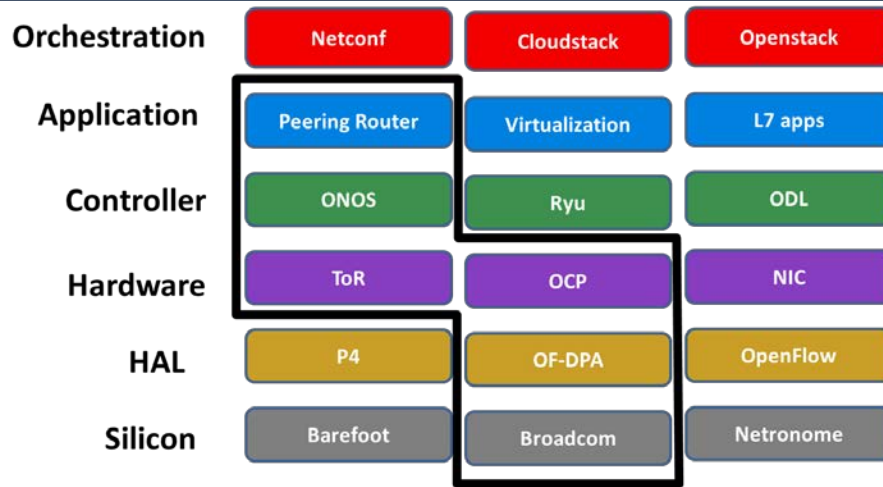


Figure 12 Solution: Vertically Integrated Stack as SDN Distribution

5.6 Human Resource – Shortage of Talents

The talents who have the knowledge of P4, ONOS, CORD, and the familiarity of open source for open networking are limited. The technical training and practice are needed. Also, the global collaboration by work together with famous organization with good reputation by sending the motivated people as interns and/or contributors. NCTU experiences working with ONF (Lab Team) in the areas of P4 and CORD Is quite successful. For this collaboration, NCTU has sent several graduate students to ONF Lab Team (US) as interns and participates in the ambassador roles and brigade team. Also, NCTU’s PhD students have participated in EURECOM (France) OpenAirInterface (OAI) open networking project and they have contribute significantly in the area of virtual Radio Access Network (vRAN) performance improvement.

6 R&D Status of P4 Switch and M-CORD at NCTU

6.1 Tested of P4 Switches and Applications Development

At NCTU, we have done the performance measurements based on the testbed of P4 switches designed by Edge-core (in Figure13, Inventec, or WNC can be the other choice) with two applications as described in Section 3. Today we open this P4 testbed to the universities and research institutes (with proper agreement) who are interested in P4 switches related research & technology exploration and P4 applications development.

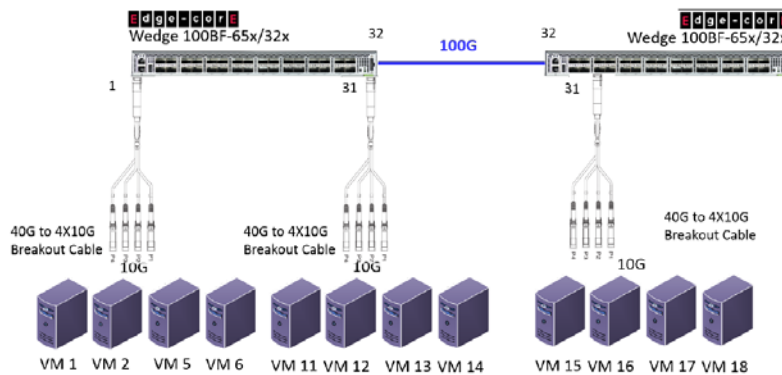


Figure 13 Topology of P4 testbed at NCTU

6.2 6.2 M-CORD architecture, Prototype, and Applications

Figure 14 shows a 4G OAI-based M-CORD Prototype at NCTU. This prototype consists of two components: one is USRP X310 (RRU+BBU) as OAI eNB, the other is CORE-in-a-box as OAI EPC with ONOS as SDN controller(MME) and P4 switches (S+P GW)+ HSS

The M-CORD-in-a-box and M-CORD Prototype at NCTU in Figure 16 is prepared to be ready for 5G UE-eNB as RAN/C-RAN.

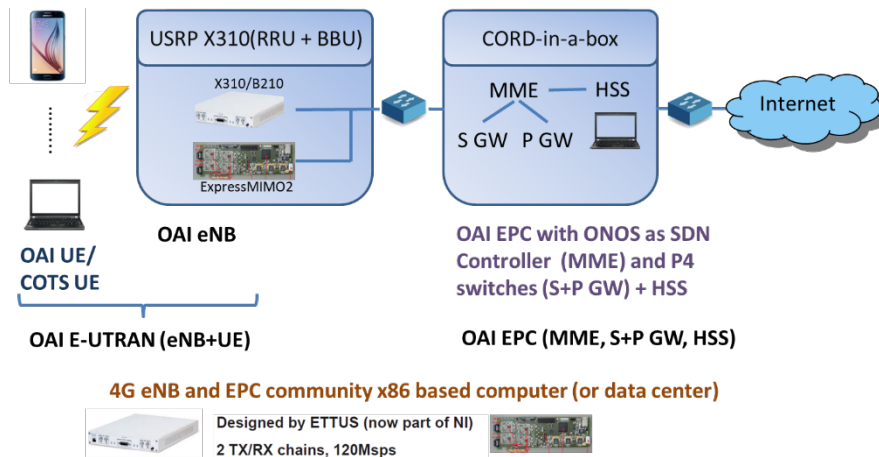


Figure 14 An OAI-based M-CORD Prototype at NCTU

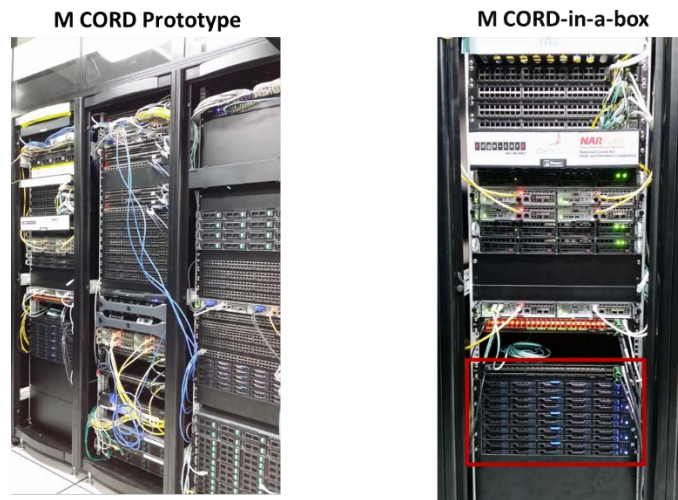


Figure 15 M-CORD-in-a-box and M CORD Prototype at NCTU

7 Conclusion

This paper addresses the discovery of the challenges/issues of network softwarization and virtualization and to find out the possible solutions. The impacts of new open networking architectures and open source, ONOS, P4 and CORD of SDN/NFV to the software defined networking industry ecosystem includes high speed network vendor (P4 switch), Rack solution provider (M-CORD box) silicon provider (SoC/IC), commodity provider (white box switch, server, storage I/O, blade)

ACKNOWLEDGEMENTS

The work as supported by the Ministry of Science and Technology (MOST) of Taiwan and National Chiao Tung University under grants: 106-2221-E-009 -008, 105-2622-8-009-008 and NCTU-ICTL 194Q707.

REFERENCES

- [1] Z. Qin, et al, "A Software defined Networking Architecture for the Internet of Things," IEEE Network Operations and Management Symposium, Krakow, Poland, 5-9 May, 2014
- [2] N. Bizanis, et al, "SDN and Virtualization Solutions for the Internet of Things," IEEE Access, Volume 2, 2016
- [3] C. Chappell, "The SDN/NFV Integration Challenge," Light Reading, Oct. 2013
- [4] J. Costa-Rquena, et al, "SDN and NFV Integration in Generalized Mobile Network Architecture," 2015 European Conference on Network and Communications (EuCNC)
- [5] J. Matias, et al, "Toward an SDN-Enabled NFV Architecture," IEEE Communications Magazine, April 2015
- [6] E. Hossain and M. Hsan, "5G Cellular: Key Enabling Technologies and Research Challenges," IEEE Instrumentation and Measurement Magazine, June, 2015
- [7] Y. Jararweh, et al, "SDIoT: a software-defined based internet of things framework," Journal of Ambient and Humanized Computing, August 20
- [8] Galis, "Challenges in 5G Network and Service Softwarization and Infrastructure," Workshop on Network Softwarization, Sep. 21, 2015, Turin
- [9] M. Corici, et al, "Unleashing the potential of virtualization by the right toolkits and open testbeds," Fraunhofer Institute FOKUS, Germany, 2015
- [10] M. Liyanage, et al, "Software Defined Mobile Networks beyond LTE Network Architecture," Wiley 2015
- [11] Cbermardo, et al, "An Architecture for software Defined Wireless Networking," IEEE Wireless Communications, June 2014
- [12] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, D. Walker, "P4: Programming Protocol-Independent Packet Processors," ACM SIGCOMM CCR, July, 2014
- [13] A. Sivaraman, C. Kim, R. Krishnamoorthy, A. Dixit, M. Budiu, "DC.p4: Programming the Forwarding Plane of a Data-Center Switch," SOSR 2015, Santa Clara, CA June, 2015
- [14] S. Signorello, R. State, J. Francois, O. Festor, "NDN.p4: Programming Information-Centric Data-Planes," 2016 European Union
- [15] Pat Bosshar, Glen Gibb, Hun-Seok Kim, George Varghese, Nick McKeown, Martin Izzard, Fernando

- Mujica, Mark Horowitz, "Forwarding Metamorphosis: Fast Programmable Match-Action Processing in Hardware for SDN," SIGCOMM'13, August 12–16, 2013, Hong Kong, China.
- [16] Barefoot Networks, "The World's Fastest Most Programmable Networks," Barefoot Networks June 15, 2016
- [17] Elisa Rojas Sanchez, "Clash of titans in SDN: OpenDaylight vs ONOS," Techday MARID, Spain, March 11, 2016
- [18] Bill Snow, "Transforming Service Provider Networks with ONOS and CORD," Open Networking Lab, 2015
- [19] Pankaj, et al, "Towards an open, distributed SDN OS," hotSDN'14, Chicago, PC, Aug. 22, 2015
- [20] Bill Snow and Guru Parulkar, "CORD – Central Office Re-architected as a Datacenter," ON.Lab Technical Report, Collaboration with AT&T, March 2016
- [21] Krish Prabhu, "Delivering a Software-based Network Infrastructure," AT&T Labs President and AT&T CTO, speaker of EE of Columbia University Engineering Oct. 15, 2015
- [22] LuongVy Le, Li-Ping Tung, Bao-Shuh Paul Lin, "SDN/NFV, Machine Learning, and Big Data Driven Network Slicing," IEEE 1st 5G World Forum, Santa Clara, CA, July 9-11,2018
- [23] World's fastest and most programmable switch series up to 6.5Tbps, Tofino by Barefoot Networks June 15, 2016
- [24] Yun-Chen Chang, "Traffic Classification for Dynamic QoS Control Based on P4-Switch," MS thesis, National Chiao Tung Univ., July 2017, Hsinchu, Taiwan
- [25] Hsueh-Tzu Liu, "Dynamic Load Balancing and Congestion Avoidance based on P4 Switch," MS thesis, National Chiao Tung Univ., July 2017, Hsinchu, Taiwan
- [26] IEEE NetSoft 2018, the 4th IEEE International Conference on Network Softwarization 2018 (NetSoft) Montreal, Canada, website: ieee.netsoft.org

Optimizing TCP Goodput and Delay in Next Generation IEEE 802.11 (ax) Devices

Oran Sharon

*Department of Computer Science
Netanya Academic College
1 University St.
Netanya 42365
Israel*

Yaron Alpert

*Texas Instruments
26 Zarchin St.
Ra'anana 4366250*

oran@math.haifa.ac.il ; Yaron@TI.com

ABSTRACT

In this paper we suggest three scheduling strategies for the IEEE 802.11ax transmission of DL unidirectional TCP data from the Access Point to stations. Two strategies are based on the Single User operation mode and one is based on the Multi User operation mode, using Multi User Multiple-Input-Multiple-Output (MU-MIMO) and OFDMA. We measure the Goodput of the system as a function of the time intervals over which these Goodputs are received in all three strategies. For up to 8 stations the MU strategy outperforms the SU. For 16 and 32 stations the SU and MU strategies perform about the same. For 64 stations the SU strategies outperform the MU significantly. We also checked the influence of the Delayed Acks feature on the received Goodputs and found that this feature has significance only when the TCP data segments are relatively short.

Keywords: 802.11ax; TCP; Aggregation; Reverse Direction; Transmission Opportunity; Goodput; MIMO; Multi User; OFDMA;

1 Introduction

1.1 Background

The latest IEEE 802.11 Standard (WiFi) [1], created and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802.11), is currently the most effective solution within the range of Wireless Local Area Networks (WLAN). Since its first release in 1997 the standard provides the basis for Wireless network products using the WiFi brand, and has since been improved upon in many ways. One of the main goals of these improvements is to increase the system throughput provided by users and to improve the standard's Quality of Service (QoS) capabilities. To fulfill the promise of increasing IEEE 802.11 performance and QoS capabilities, shall be a new amendment (IEEE 802.11ax - also known as High Efficiency (HE))

) was recently introduced [2]. IEEE 802.11ax is considered to be the sixth generation of a WLAN in the IEEE 802.11 set of WLAN types and is a successor to IEEE 802.11ac [3, 4]. The scope of the IEEE 802.11ax amendment is to define modifications for both the IEEE 802.11 PHY and MAC layers that enable at least four-fold improvement in the average throughput per station in densely deployed networks [5–8]. Currently IEEE 802.11ax project is finalizing revision 2.0, which will be the baseline for WFA IEEE 802.11ax certification.

1.2 Research question

In order to achieve its goals, one of the main challenges of IEEE 802.11ax is to enable UL and DL simultaneous transmissions by several stations and to improve Quality-of-Service performance. The current paper is a continuation to papers [9–11]. In these papers the authors suggest scheduling strategies for the parallel transmissions of the AP to a given set of stations using new features of IEEE 802.11ax. The authors assume UDP-like traffic where the AP transmits data MSDUs to the stations, which reply with MAC acknowledgments. In this paper we assume a DL unidirectional TCP-like traffic in which the AP transmits TCP Data MSDUs to a given set of stations, and the stations reply with TCP Ack MSDUs. As far as we know the issue of transmitting TCP traffic over IEEE 802.11ax has not yet been investigated. We suggest several scheduling strategies for the transmissions of TCP traffic over the DL using Single User (SU) and Multi User (MU) modes for 1, 4, 8, 16, 32 and 64 stations scenarios over a reliable channel. This is one of the aspects to compare between new amendments of the IEEE 802.11 standard [12]. In this paper we are interested in finding an upper bound on the maximum DL unidirectional TCP Goodput that can be achieved by IEEE 802.11ax and comparing between the various scheduling strategies. Therefore, we assume the traffic saturation model where TCP connections always have data to transmit and the TCP Ack is generated immediately by receivers. Second, we neutralize any aspects of the PHY layer as the number of Spatial Streams (SS) in use and channel correlation when using Multi User Multiple Input Multiple Output (MU-MIMO), the use in the sounding protocol etc.

As mentioned, we assume that every TCP connection has an unlimited number of TCP Data segments to transmit, and we assume that transmissions are made using an optimized (in terms of overhead reduction) two level aggregation scheme to be described later. Our goal is to find an upper bound on the maximum possible Goodput that the wireless channel enables the TCP connections, where the TCP itself does not impose any limitations on the offered load, i.e. on the rate that MSDUs are given for transmission to the MAC layer of the IEEE 802.11ax. We also assume that the AP and the stations are the end points of the TCP connections. Following e.g. [13–16] it is quite common to consider short Round Trip Times (RTT) in this kind of high speed network such that no retransmission timeouts occur. Moreover, we assume that every TCP connections' Transmission Window can always provide as many MSDUs to transmit as the IEEE 802.11ax protocol limits enable. This assumption follows the observation that aggregation is useful in a scenario where the offered load on the channel is high. Finally, we assume that every TCP Ack either acknowledges one TCP Data segment, or it acknowledges two TCP Data segments. The latter possibility is denoted Delayed Acks, a feature in TCP that enables a TCP Ack to acknowledge two TCP Data segments.

This research is only a first step in investigating TCP traffic in IEEE 802.11ax. In our further papers we plan to address other TCP traffic scenarios to investigate such as UL unidirectional TCP traffic and bi-directional TCP traffic.

1.3 Previous works

The issue of TCP traffic over IEEE 802.11ax that involves bidirectional data packet exchange has not yet been studied. Most of the research papers on IEEE 802.11ax thus far examine different access methods to enable efficient multi-user access to random sets of stations. For example, in [17] the authors deal with the introduction of Orthogonal Frequency Division Multiple Access (OFDMA) into IEEE 802.11ax to enable multi user access. They introduce an OFDMA based multiple access protocol, denoted Orthogonal MAC for IEEE 802.11ax (OMAX), to solve synchronization problems and reduce overhead associated with using OFDMA. In [18] the authors suggest an access protocol over the UL of an IEEE 802.11ax WLAN based on MU-MIMO and OFDMA PHY. In [19] the authors suggest a centralized medium access protocol for the UL of IEEE 802.11ax in order to efficiently use the transmission resources. In this protocol, stations transmit requests for frequency sub-carriers, denoted Resource Units (RU), to the AP over the UL. The AP allocates RUs to the stations which later use them for data transmissions over the UL. In [20] a new method to use OFDMA over the UL is suggested, where MAC Protocol Data Units (MPDU) from the stations are of different lengths. In [21–24] a new version of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol, denoted Enhanced CSMA/CA (CSMA/ECA) is suggested for MU transmissions, which is suitable for IEEE 802.11ax . A deterministic BackOff is used after a successful transmission, and the BackOff stage is not reset after service. The BackOff stage is reset only when a station does not have any further MPDUs to transmit. CSMA/ECA enables more efficient use of the channel and enhanced fairness. In [25] the authors assume a network with legacy and IEEE 802.11ax stations and examine fairness issues between the two sets of stations.

We would like to mention that the issue of TCP traffic over IEEE 802.11ac networks (the predecessor standard of IEEE 802.11ax) has already been investigated, e.g. in [26–28], for DL TCP traffic, UL TCP traffic and both DL and UL TCP traffic. However, in all these works there is no possibility of using the MU operation mode over the UL, a feature that was first introduced in IEEE 802.11ax .

The remainder of the paper is organized as follows: In Section 2 we describe the new mechanisms of IEEE 802.11ax relevant to this paper. In Section 3 we describe the scheduling strategies that we suggest in SU and MU modes. We assume the reader is familiar with the basics of PHY and MAC layers of IEEE 802.11 described in previous papers, e.g. [29]. In Section 4 we analytically compute the Goodputs of the various scheduling strategies. In Section 5 we present the Goodputs of the various scheduling strategies and Section 6 summarizes the paper. In the Appendix we show how to efficiently schedule MPDUs in the various scheduling strategies. Lastly, moving forward, we denote IEEE 802.11ax by 11ax .

2 The new features in IEEE 802.11ax

IEEE 802.11ax focuses on implementing mechanisms to efficiently serve more users, enabling consistent and reliable streams of data (average throughput per user) in the presence of multiple users. In order to meet these targets 11ax addresses several new mechanisms in both the PHY and MAC layers. At the PHY layer, 11ax enables larger OFDM FFT sizes (4X larger) and therefore every OFDM symbol is $12.8\mu\text{s}$ compared to $3.2\mu\text{s}$ in IEEE 802.11ac, the predecessor of 11ax . By narrower sub-carrier spacing (4X closer) the protocol efficiency is increased because the same Guard Interval (GI) is used both in 11ax and in previous versions of the standard.

In addition, to increase the average throughput per user in high-density scenarios, 11ax introduces two new Modulation Coding Schemes (MCSs), MCS10 (1024 QAM) and MCS 11 (1024 QAM 5/6), applicable for transmission with bandwidth larger than 20 MHz.

In this paper we use the Transmission Opportunity (TXOP) feature first introduced in IEEE 802.11n [30]. This feature allows a station, after gaining access to the channel, to transmit several PHY Protocol Data Units (PPDUs) in a row without interruption, and can also allocate some of the TXOP time interval to one or more receivers in order to allow data transmission in the reverse link. This is termed Reverse Direction (RD). For scenarios with bidirectional traffic such as TCP Data segments/Ack segments, this approach is very efficient as it reduces contention in the wireless channel.

We focus on optimizing the TXOP duration and pattern, PPDU duration and the 11ax's two-level aggregation scheme working point first introduced in IEEE 802.11n [30], in which several MPDUs can be aggregated to be transmitted in a single PHY Service Data Unit (PSDU). Such aggregated PSDU is denoted Aggregate MAC Protocol Data Unit (A-MPDU) frame. In two-level aggregation every MPDU can contain several MAC Service Data Units (MSDU). MPDUs are separated by an MPDU Delimiter field of 4 bytes and each MPDU contains MAC Header and Frame Control Sequence (FCS) fields. MSDUs within an MPDU are separated by a SubHeader field of 14 bytes. Every MSDU is rounded to an integral multiple of 4 bytes together with the SubHeader field. Every MPDU is also rounded to an integral multiple of 4 bytes.

In 11ax the size of an MPDU is limited to 11454 bytes and the size of the A-MPDU frame is limited to 4,194,304 bytes. The transmission time of the PPDU (PSDU and its preamble) is limited to 5.484ms (5484 μ s) due to the L-SIG (one of the legacy preamble's fields) duration limit [1]. The A-MPDU frame structure in two-level aggregation is shown in Figure 1.

IEEE 802.11ax also enables extension of the acknowledgment mechanism by using an acknowledgment window of 256 MPDUs. In this paper we also assume that all MPDUs transmitted in an A-MPDU frame are from the same Traffic Stream (TS). In this case up to 256 MPDUs are allowed in an A-MPDU frame of 11ax.

Finally, in 11ax it is possible to transmit/receive simultaneously to/from up to 74 stations over the DL/UL respectively using MU.

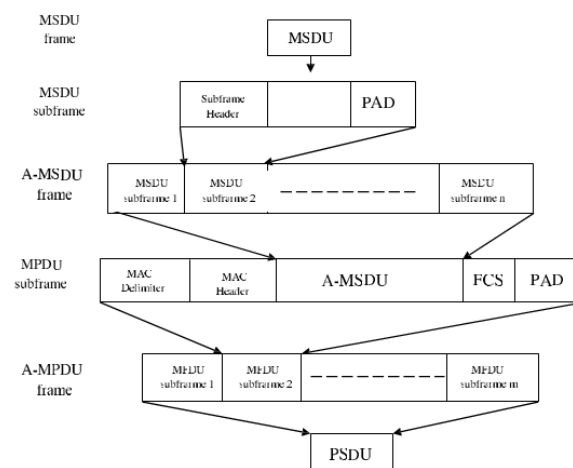


Figure 1: The generation of an A-MPDU frame in two-level aggregation

3 Model

3.1 HE scheduling strategies for TCP Usage

We compare between 11ax contention based Single User (SU), Reverse Direction (RD) SU and Multi User (MU) TCP DL unidirectional scheduling strategies in order to optimize the performance of DL single direction TCP connections, from the AP to stations.

3.1.1 Scheduling strategy 1 - HE DL Single User Reverse Direction unidirectional TCP

Recall that Reverse Direction (RD) is a mechanism by which the owner of a Transmission Opportunity (TXOP), the AP in our case, can enable its receiver to immediately transmit back the TCP Acks during the TXOP so that the receiver does not need to initiate UL transmission by using the Extended Distributed Coordination Function (EDCF) channel access method defined in IEEE 802.11e [1]. This is particularly efficient for bi-directional traffic such as TCP Data/Ack segments as it reduces overhead caused by collisions.

We examine a HE RD based scheduling strategy in which the AP transmits DL HE SU A-MPDU frames containing MPDUs of TCP Data segments to a station and enables the station to answer with an UL HE SU A-MPDU frame containing MPDUs frames of TCP Acks segments. Both the AP and the stations apply the two-level aggregation. We assume the following scenario to use RD, as is illustrated in Figure 2.

After waiting AIFS and BackOff according to the 802.11 air access EDCA procedure, the AP initiates a TXOP by transmitting n DL HE SU A-MPDU frames in a row. Every such DL PDU transmission, followed by receiving the BACK frame from the station, is denoted a HE DL RD TCP Data cycle. In its last DL HE SU A-MPDU frame the AP sets the RDG bit [1], enabling the station to respond with an UL HE SU A-MPDU frame containing TCP Ack segments. The AP then responds with a BACK frame and terminates the TXOP with the CF-End frame [1]. The transmission of the UL HE SU A-MPDU frame by the station, followed by the BACK transmission from the AP, is denoted a HE UL RD TCP Ack cycle.

In this HE RD based scheduling strategy we assume that there are no collisions and TXOP are repeated over the channel one after the other. This is made possible by configuring the stations in a way that prevents collisions. For example, the stations are configured to choose their BackOff intervals from very large contention intervals, other than the default ones [1]. Thus, the AP always wins over the channel without collisions.

In the case where the AP maintains TCP connections with S stations in parallel, it transmits to the stations using Round Robin i.e. , after maintaining a TXOP with a station the AP initiates a TXOP with the next station and so on.

3.1.2 Scheduling strategy 2 - HE DL Single User contention based unidirectional TCP

This HE SU scheduling strategy is shown in Figure 3. In this strategy the AP uses TXOPs but not RD: when the AP gets access to the channel it transmits DL HE SU A-MPDU frames containing TCP Data segments to a station in a row. Every transmission of a single DL HE SU A-MPDU frame from the AP is followed by a BACK frame transmission from the destination station; see Figure 3(A).

In this scheduling strategy both the AP and the stations contend in parallel for accessing the air channel in every transmission attempt, using the EDCF channel access method. In

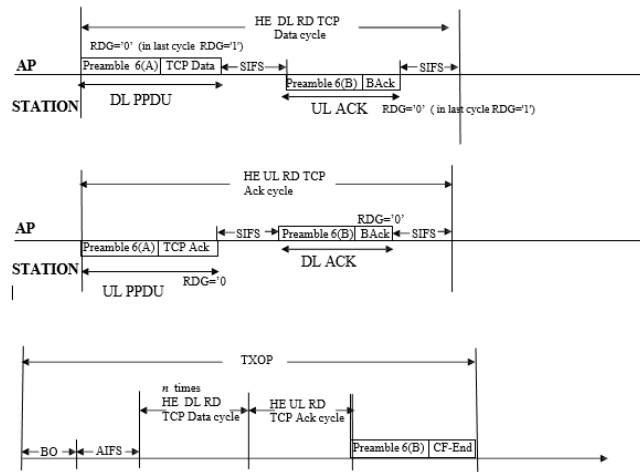
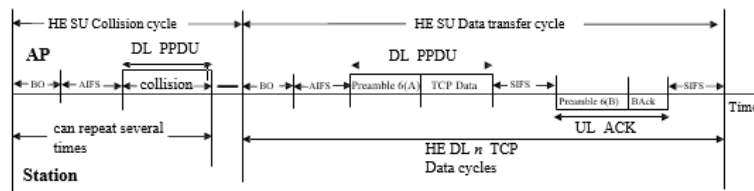
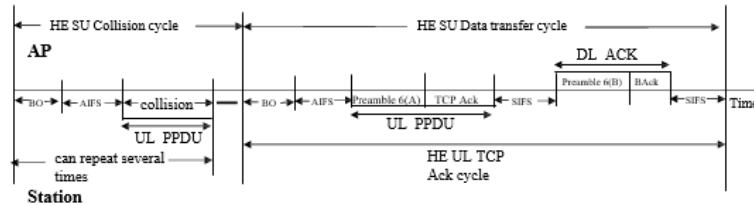


Figure 2: Scheduling strategy 1: the scheduling strategy that uses HE Reverse Direction.



(A) AP transmission when the AP wins channel access



(B) Station transmission when a station wins channel access

Figure 3: Scheduling strategy 2: the contention based scheduling strategy

case the AP fails to gain access to the channel during its first attempt, it tries to access the channel again according to EDCF, with re-try penalty (longer BackOff interval) as shown in Figure 3(A).

The AP transmits to the stations in a Round Robin fashion. After transmitting TCP Data segments to a station, the AP does not serve that station again before receiving TCP Ack segments from the station and before the AP returns again to the station in the Round Robin order. Notice from the above that if the AP returns to a station in the Round Robin order before that station transmits TCP Ack segments to the AP, the AP skips over the station.

A station transmits to the AP only when it has TCP Ack segments, and it transmits the TCP Acks in one UL HE SU A-MPDU frame. See Figure 3(B).

3.1.3 Scheduling strategy 3 - HE DL simultaneous Multi User unidirectional TCP

In the HE DL unidirectional TCP Multi User mode the AP transmits TCP Data to and receives TCP Acks from several stations in parallel. We assume the following DL unidirectional TCP where simultaneous DL TCP Data is sent by the AP to multiple stations in the same PPDU and the TCP Acks are sent simultaneously by the stations at the same TXOP by using Multi User, as is illustrated in Figure 4.

In this HE DL MU scheduling strategy, after waiting the BackOff and AIFS intervals, the AP receives an air access and starts a TXOP by transmitting n DL HE MU A-MPDU frames containing TCP Data segments to a group of stations simultaneously. In every DL HE MU A-MPDU frame the AP transmits to a different set of stations in the group. After receiving the UL HE MU BAcK frames from the group of stations simultaneously, the so-called HE DL MU TCP Data cycle ends and such a cycle can now repeat itself several times.

In order to transmit to a group of stations simultaneously, the AP allocates Resource Units (RU), i.e. sub-channels, per served station. RU allocation is done at the DL for TCP Data segments and at the UL for the TCP Acks. The AP signals the stations when and how to transmit, i.e. their UL RU allocation using one of two possible methods. In the first method the AP transmits a unicast Trigger Frame (TF) to every station that contains the UL RU allocation. This frame is a control MPDU frame that is added to the other Data MPDUs that the AP transmits to a station in a DL HE MU A-MPDU frame. The alternative method is to add an HE Control Element to every MPDU in the DL HE MU A-MPDU frame that is transmitted to every station. In the following Goodput computations we optimize the amount of overhead used due to the above methods by computing the minimum overhead needed as a function of the number of data MPDUs in the DL HE MU A-MPDU frame.

At the end of the last HE DL MU TCP Data cycle the AP initiates a HE UL MU TCP Ack cycle by transmitting the broadcast Trigger Frame (TF). This TF solicits TCP Ack transmissions from the stations to the AP. At this transmission the stations transmit TCP Ack segments using UL HE MU A-MPDU frames. Every station transmits its TCP Ack segments in a different UL HE MU A-MPDU frame. The AP acknowledges the stations' UL HE MU A-MPDU frames by generating and transmitting a single DL Multi Station BAcK frame. At this stage the HE UL MU TCP Ack cycle ends and a new series of HE DL MU TCP Data cycle(s) and HE MU UL TCP Ack cycle begin.

As in the SU RD based scheduling strategy we assume that there are no collisions by increasing the size of the congestion interval from which the stations choose their EDCF BackOff extended interval.

3.2 IEEE 802.11 Frames/PPDU formats

In Figure 5 we show the 802.11 frames' formats of the BAcK, Multi Station BAcK, TF and CF-End frames used in the various scheduling strategies. In Figure 6 we show the various PPDUs' formats used in the various scheduling strategies shown in Figures 2-4.

For the TCP Data/Ack segments' transmission in Figure 2, scheduling strategy 1, the PPDU format in Figure 6(A) is used while the BAcK and CF-End frames are transmitted using the legacy mode in Figure 6(B).

For the TCP Data/Ack segments' transmission in Figure 3, scheduling strategy 2, the PPDU in Figure 6(A) is used while the BAcK frames are transmitted by the legacy mode shown in Figure 6(B).

For the TCP DL Data segments' transmission in Figure 4, scheduling strategy 3, the PPDU format in Figure 6(C) is used and the BACKs are transmitted using the PPDU format in Figure 6(D). The TCP UL Ack segments are transmitted by the PPDU format in Figure 6(D) and the TF and DL Multi Station Ack frames are transmitted by the legacy PPDU format in Figure 6(B).

In the 11ax PPDU formats we find the HE-LTF fields, the number of which equals the number of SSs in use; 4 in our case. In this paper we assume that each such field is composed of 2X LTF and therefore of duration $7.2\mu\text{s}$ [2].

Notice also that the PSDU frame in 11ax contains a Packet Extension (PE) field. This field is mainly used in MU mode and we assume that it is $0\mu\text{s}$ in SU and the longest possible in MU, $16\mu\text{s}$.

In the HE-SIG-B field used in the PPDU format of Figure 6(C) the Modulation/Coding Scheme (MCS) that is used for this field is the minimum between MCS4 and the one used for the data transmissions [2]. The length of this field is also a function of the number of stations to which the AP transmits simultaneously. Therefore, in the case of 4 stations for example, the HE-SIG-B field duration is $8\mu\text{s}$ for MCS0 and MCS1 and $4\mu\text{s}$ for MCS2-4 following section 23.3.9.8 in [2]. For MCS5-MCS11 it is $4\mu\text{s}$ as for MCS4.

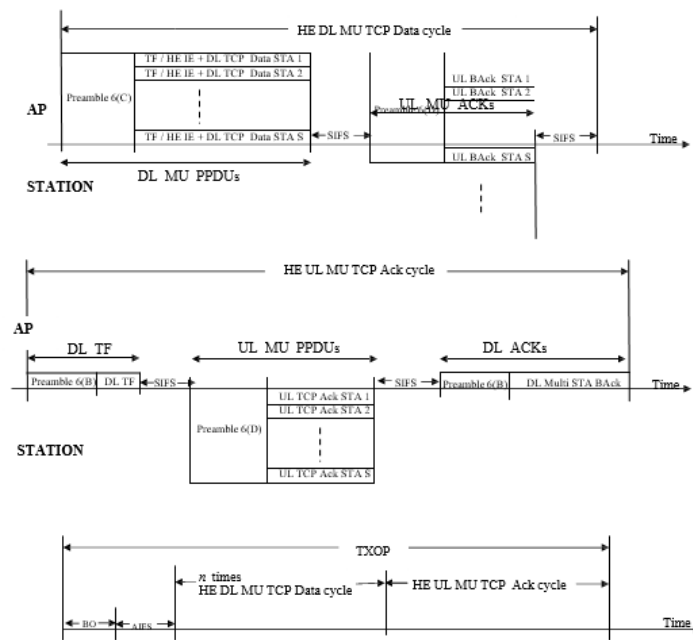


Figure 4: Scheduling strategy 3: the HE Multi User scheduling strategy

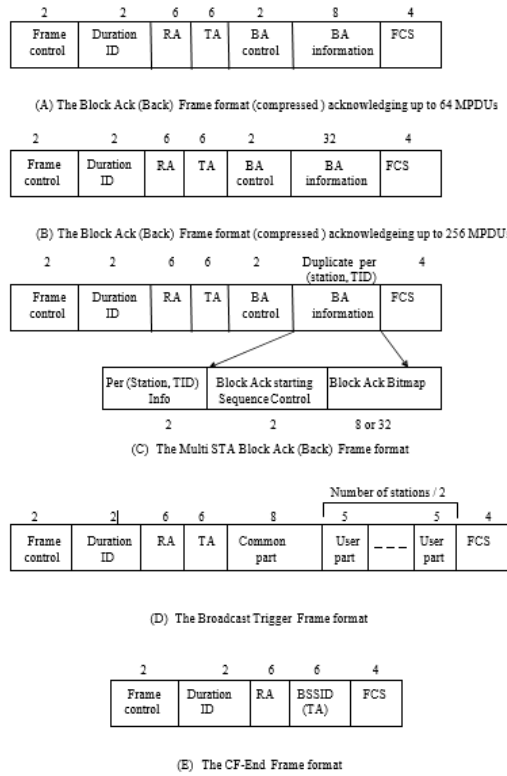


Figure 5: The Back, Multi Station Back, Trigger Frame and CF-End frames' formats.

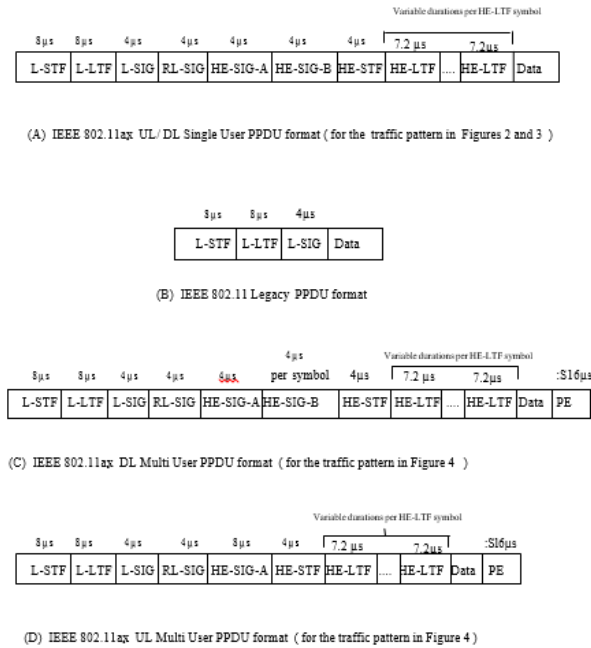


Figure 6: The PPDUs' formats in the SU and MU modes.

Table 1: The PHY rates and preambles in IEEE 802.11ax used in Single User mode and in scheduling strategies 1 and 2. A 160 MHz channel is assumed, with 4 Spatial Streams. The BAck frames are conducted at the basic rates' set.

MCS	1		2	
	SU UL/DL data transmission rate		SU UL/DL BAck transmission rate	
	PHY Rate (Mbps) GI= 0.8 μ s	Preamble (μ s)	PHY Rate (legacy) (Mbps) GI= 0.8 μ s	Preamble (μ s)
0	288.2	64.8	48.0	20.0
1	576.5	64.8	48.0	20.0
2	864.7	64.8	48.0	20.0
3	1152.9	64.8	48.0	20.0
4	1729.4	64.8	48.0	20.0
5	2305.9	64.8	48.0	20.0
6	2594.1	64.8	48.0	20.0
7	2882.4	64.8	48.0	20.0
8	3458.8	64.8	48.0	20.0
9	3848.1	64.8	48.0	20.0
10	4323.5	64.8	48.0	20.0
11	4803.9	64.8	48.0	20.0

3.3 Parameters' values

We assume the 5GHz band, a 160MHz channel and that the AP and each station has 4 antennae. In SU mode, i.e. in scheduling strategies 1 and 2, the AP and the stations use up to 4 Spatial Streams and the entire channel is devoted to transmissions of the AP and stations. The BAck frames are transmitted using legacy mode and the basic rates' set is used. The PHY rate R_{legacy} is set to the largest basic rate that is smaller or equal to the TCP Data/Ack segments' transmission rate R_{TCP} .

In Table 1 we show the PHY rates and the length of preambles used in SU mode in scheduling strategies 1 and 2 and in the various MCSs. The values are taken from [2].

When using MU mode in scheduling strategy 3, the 160MHz channel is divided into $S/4$ channels of $160 \cdot 4/S$ MHz each, $S = 4, 8, 16, 32, 64$. When $S = 4$ the 160MHz is used in MU-MIMO. For $S > 4$ MU-MIMO+OFDMA is used. The AP transmits over the DL to 4 stations in every such channel, by allocating a single Spatial Stream per station i.e. in every channel 4 Spatial Streams are allocated. For example, for $S = 64$ there are 16 channels of 10MHz each; in each one the AP transmits to 4 stations. The stations transmit to the AP over the UL in a symmetrical way to that of the AP over the DL.

The TF and the Multi Station BAck frames are transmitted using the legacy mode and the PHY rate R_{legacy} is set to the largest basic rate that is smaller or equal to the TCP Data/Ack segments' transmission rate R_{TCP} . The minimal basic PHY rate is 6Mbps. In the case of R_{TCP} smaller than 6Mbps, R_{legacy} is never less than 6Mbps. This can occur in the case of 64 stations.

In Table 2 we show the PHY rates and the preambles used in scheduling strategy 3, in the various MCSs and in all cases of the number of stations S , i.e. $S = 4, 8, 16, 32$ and 64.

We assume the Best Effort Access Category in which $AIFS = 43\mu s$ for the AP and $52\mu s$ for a station, $SIFS = 16\mu s$ and $CW_{min} = 16$ for the transmissions of the AP. Recall that in scheduling strategies 1 and 3 we assume there are no collisions between the AP and the stations. The BackOff interval is a random number chosen uniformly from the range

$[0, \dots, CW_{min} - 1]$. Since we consider a very 'large' number of transmissions from the AP in scheduling strategies 1 and 3, we take the BackOff average value of $\frac{CW_{min}-1}{2}$, and the average BackOff interval for the AP is $\frac{CW_{min}-1}{2}$ SlotTime which equals $67.5\mu s$ for a SlotTime = $9\mu s$.

Concerning the transmission in non-legacy mode, an OFDM symbol is $12.8\mu s$. In the DL we assume a GI of $0.8\mu s$ and therefore the symbol in this direction is $13.6\mu s$. In the UL MU we assume a GI of $1.6\mu s$ and therefore the symbol in this direction is $14.4\mu s$. The UL GI is $1.6\mu s$ due to UL arrival time variants. In UL SU the GI is $0.8\mu s$. When considering transmissions in legacy mode, the symbol is $4\mu s$ containing a GI of $0.8\mu s$.

We assume that the MAC Header field is of 28 bytes and the Frame Control Sequence (FCS) field is of 4 bytes. Finally, we assume TCP Data segments of $L_{DATA} = 1460, 464$ and 208 bytes. Therefore, the resulting MSDUs' lengths are $L_{DATA} = 1508, 512$ and 256 bytes respectively (20 bytes of TCP Header plus 20 bytes of IP Header plus 8 bytes of LLC SNAP are added). Together with the Sub Header field and rounding to an integral multiple of 4 bytes, every MSDU is now of $LenD = 1524, 528$ and 272 bytes respectively. Due to the limit of 11454 bytes on the MPDU size, 7, 21 and 42 such MSDUs are possible respectively in one MPDU. The TCP receiver transmits TCP Acks. Every MSDU containing a TCP Ack is of $L_{Ack} = 48$ bytes (20 bytes of TCP Header + 20 bytes of IP Header + 8 bytes of LLC SNAP). Adding 14 bytes of the SubHeader field and rounding to an integral multiple of 4 bytes, every MSDU is of $LenA = 64$ bytes, and every single MPDU, again due to the size limit of 11454 bytes, can contain up to 178 MSDUs (TCP Acks). Thus, the receiver can transmit up to $N_{MAX} = 256 \cdot 178$ TCP Acks in a single HE UL A-MPDU frame.

Table 2: The PHY rates and preambles in IEEE 802.11ax in scheduling strategy 3. A 160 MHz channel is assumed, with 4 Spatial Streams. The TF and Back transmissions are conducted at the basic rate set.

MCS	1		2		3	
	MU UL data transmission rate		MU DL data transmission rate		DL TF/Multi Station BAck transmission rate	
	PHY Rate (Mbps per 1 SS) GI= $1.6\mu s$	Preamble (μs)	PHY Rate (Mbps per 1 SS) GI= $0.8\mu s$	Preamble (μs)	PHY Rate (legacy) (Mbps) GI= $0.8\mu s$	Preamble (μs)
4 stations						
0	68.1	64.8	72.1	72.8	48.0	20.0
1	136.1	64.8	144.1	72.8	48.0	20.0
2	204.2	64.8	216.2	68.8	48.0	20.0
3	272.2	64.8	288.2	68.8	48.0	20.0
4	408.3	64.8	432.4	68.8	48.0	20.0
5	544.4	64.8	576.5	68.8	48.0	20.0
6	612.5	64.8	648.5	68.8	48.0	20.0
7	680.6	64.8	720.6	68.8	48.0	20.0
8	816.7	64.8	864.7	68.8	48.0	20.0
9	907.4	64.8	960.7	68.8	48.0	20.0
10	1020.8	64.8	1080.4	68.8	48.0	20.0
11	1134.2	64.8	1201.0	68.8	48.0	20.0
8 stations						
0	34.0	64.8	36.0	76.8	36.0	20.0
1	68.1	64.8	72.1	76.8	48.0	20.0
2	102.1	64.8	108.1	72.8	48.0	20.0
3	136.1	64.8	144.1	72.8	48.0	20.0
4	204.2	64.8	216.2	68.8	48.0	20.0
5	272.2	64.8	288.2	68.8	48.0	20.0
6	306.3	64.8	324.3	68.8	48.0	20.0
7	340.3	64.8	360.3	68.8	48.0	20.0
8	408.3	64.8	432.4	68.8	48.0	20.0
9	453.7	64.8	480.4	68.8	48.0	20.0
10	510.4	64.8	540.4	68.8	48.0	20.0

11	567.1	64.8	600.4	68.8	48.0	20.0
16 stations						
0	16.3	64.8	17.2	84.8	12.0	20.0
1	32.5	64.8	34.4	84.8	12.0	20.0
2	48.8	64.8	51.6	76.8	24.0	20.0
3	65.0	64.8	68.8	76.8	48.0	20.0
4	97.5	64.8	103.2	72.8	48.0	20.0
5	130.0	64.8	137.6	72.8	48.0	20.0
6	146.3	64.8	154.9	72.8	48.0	20.0
7	162.5	64.8	172.1	72.8	48.0	20.0
8	195.0	64.8	206.5	72.8	48.0	20.0
9	216.7	64.8	229.4	72.8	48.0	20.0
10	243.8	64.8	258.1	72.8	48.0	20.0
11	270.8	64.8	286.8	72.8	48.0	20.0

Table 2: (cont.)

MCS	1 MU UL data transmission rate		2 MU DL data transmission rate		3 DL TF/Multi Station BAck transmission rate	
	PHY Rate (Mbps per 1 SS) GI= 1.6 μ s	Preamble (μ s)	PHY Rate (Mbps per 1 SS) GI= 0.8 μ s	Preamble (μ s)	PHY Rate (legacy) (Mbps) GI= 0.8 μ s	Preamble (μ s)
32 stations						
0	8.1	64.8	8.6	104.8	6.0	20.0
1	16.3	64.8	17.2	104.8	12.0	20.0
2	24.4	64.8	25.8	84.8	24.0	20.0
3	32.5	64.8	34.4	84.8	24.0	20.0
4	48.8	64.8	51.6	80.8	48.0	20.0
5	65.0	64.8	68.8	80.8	48.0	20.0
6	73.1	64.8	77.4	80.8	48.0	20.0
7	81.3	64.8	86.0	80.8	48.0	20.0
8	97.5	64.8	103.2	80.8	48.0	20.0
9	108.3	64.8	114.7	80.8	48.0	20.0
10	121.9	64.8	129.0	80.8	48.0	20.0
11	135.4	64.8	143.4	80.8	48.0	20.0
64 stations						
0	3.5	64.8	3.8	136.8	6.0	20.0
1	7.1	64.8	7.5	136.8	6.0	20.0
2	10.6	64.8	11.3	100.8	9.0	20.0
3	14.2	64.8	15.0	100.8	12.0	20.0
4	21.3	64.8	22.5	88.8	18.0	20.0
5	28.3	64.8	30.0	88.8	24.0	20.0
6	31.9	64.8	33.8	88.8	24.0	20.0
7	35.4	64.8	37.5	88.8	24.0	20.0
8	42.5	64.8	45.0	88.8	36.0	20.0
9	47.2	64.8	50.0	88.8	36.0	20.0
10	N/A	N/A	N/A	N/A	N/A	N/A
11	N/A	N/A	N/A	N/A	N/A	N/A

4 Goodput analysis

The system Goodput analysis has two targets. The first is to find the optimal working point for each of the proposed scheduling strategies, i.e. finding the working point that maximizes the Goodput of a TXOP. By an optimal working point we refer to the number of DL TCP Data MSDUs to transmit in a TXOP, how many HE DL TCP Data cycles to transmit in a TXOP, and the optimal HE DL A-MPDU structure within each HE DL TCP Data cycle i.e. the number of MPDUs in the HE DL A-MPDU and the number of MSDUs within every MPDU.

Notice that the Goodput computed is actually the system *TCP Goodput*, i.e. the average number of TCP Data bits that are transmitted in the system per time unit. However, in SU, scheduling strategies 1 and 2, when S stations are served by the AP, a single station enjoys a given TCP Goodput G in every S^{th} TXOP only. The system provides a TCP Goodput G to *all* the stations

over S TXOPs. In the MU strategy, scheduling strategy 3, the TCP Goodput G of a TXOP is that provided to all the stations together over one TXOP.

The second target of the analysis is to find the time intervals over which the system enables a given TCP Goodput G to all of its stations. A scheduling strategy that enables a given TCP Goodput to all stations over shorter time intervals is more efficient.

4.1 Maximum Goodput of a TXOP

Computing the optimal working point per scheduling strategy, i.e. the one that maximizes the Goodput of a TXOP, is done in 3 stages:

1. The number of TCP Data segments that can be transmitted in a TXOP is limited by the number $N_{MAX} = 256 \cdot 178$ TCP Ack segments that can be transmitted in one HE UL A-MPDU frame. The number 256 is the Block Ack window size and 178 is the number of TCP Ack segments possible in one MPDU. N_{MAX} was computed in Section 3.3 and if more than N_{MAX} TCP Data segments are transmitted in a TXOP, stations begin to accumulate TCP Acks and the Goodput decreases.
2. In order to maximize the Goodput the TCP Acks shall be transmitted in the minimal possible number of MPDUs in order to minimize overhead associated with MPDUs containing TCP Acks. For N TCP Acks, $1 \leq N \leq N_{MAX}$ the number of MPDUs is $\lceil \frac{N}{178} \rceil$ 'See Section 3.3'.
3. For every number N of TCP Data segments transmitted in a TXOP, $1 \leq N \leq N_{MAX}$, it is necessary to determine what is the number of HE DL A-MPDU frames for transmission in a TXOP, and the structure of each HE DL A-MPDU frame, i.e. how many MPDUs are in every HE DL A-MPDU frame and how many MSDUs are in every MPDU. This is necessary in order to minimize overhead associated with MPDU and HE DL A-MPDU frames that contain TCP Data segments. This computation is carried on in the Appendix.

For scheduling strategies 1 and 3 we provide a mathematical analysis to compute the Goodput of a TXOP. This analysis was verified by the NS3 simulator. The analysis and simulation results match perfectly. This is not surprising as there is no stochastic process in these strategies. Therefore, we later omit their simulation results. For scheduling strategy 2 we only use simulation to compute the Goodput.

4.1.1 Goodput analysis for scheduling strategy 1 - HE DL Single User Reverse Direction unidirectional TCP

For this HE SU RD scheduling strategy notice that $N_{MAX} = 256 \cdot 178$ TCP Acks that a station can transmit in one HE UL RD TCP Ack cycle is an upper bound on the number of DL TCP Data MSDUs that can be transmitted by the AP in a TXOP. Using a larger

number will cause the receiver to accumulate TCP Acks and the Goodput to decrease. See Section 3.3.

An HE SU RD TXOP has a fixed overhead of the AIFS and BackOff intervals, and the transmission time of the CF-End frame and its associated preamble 6(B). In addition there are overheads associated with the transmission of an MPDU frame and a HE DL SU A-MPDU frame. The

MPDU's overhead is composed of the MAC Header, MPDU Delimiter and the FCS fields. The HE DL SU A-MPDU overhead for scheduling strategy 1

is $Pr(6(A)) + Pr(6(B)) + T(BAck) + 2 \cdot SIFS$. See Figure 2 .

Due to the various overheads it may not always be worthwhile to transmit N_{MAX} TCP Data MSDUs in a TXOP. For instance, a single MSDU can cause the creation of both a new HE DL SU A-MPDU and an MPDU with all of the associated overhead that overall can reduce the Goodput.

Assume that N TCP Data MSDUs are transmitted in a TXOP. For a reliable channel it is clearly most efficient to transmit the N TCP Ack MSDUs in the smallest number of MPDUs i.e. in $\lceil \frac{N}{178} \rceil$ MPDUs. This will reduce the overhead associated with MPDUs containing TCP Ack MSDUs to a minimum. Recall that all the TCP Ack MSDUs are transmitted in one HE UL SU A-MPDU frame.

Recall now that in the Appendix we show how to schedule N TCP Data MSDUs in the most efficient way in a TXOP, i.e. the scheduling that results in the smallest HE UL SU A-MPDUs' and MPDUs' overheads. Assume that the optimal scheduling of N TCP Data MSDUs is in n HE DL SU A-MPDUs. Let A_i and N_i be the numbers of MPDUs and MSDUs respectively in HE DL SU A-MPDU number i , $1 \leq i \leq n$.

Let AIFS and BO denote the length, in μs , of the AIFS and average BackOff intervals. Let O_M be the total length, in bytes, of the MAC Header, MPDU Delimiter and FCS fields. The Goodput of scheduling strategy 1 is given by Eq. 1, assuming the AP transmits N TCP Data MSDUs in n HE DL SU RD TCP Data cycles:

$$Goodput = \frac{N \cdot (L_{DATA} \cdot 8)}{AIFS + BO(average) + \sum_{i=1}^n T_{Data}^{Cycle}(A_i, N_i) + T_{Ack}^{Cycle} + Pr(6(B)) + T(CF - End)} \quad (1)$$

Where:

$$T_{Data}^{cycle}(A_i, N_i) = Pr(6(A)) + TSym_{DL} \cdot \left\lceil \frac{(N_i \cdot Len^D + A_i \cdot O_M) \cdot 8 + 22}{TSym_{DL} \cdot R_{DL}} \right\rceil + Pr(6(B)) + T(BAck) + 2 \cdot SIFS$$

$$T_{Ack}^{cycle} = Pr(6(A)) + TSym_{UL} \cdot \left\lceil \frac{(N \cdot Len^A + \lceil \frac{N}{178} \rceil \cdot O_M) \cdot 8 + 22}{TSym_{UL} \cdot R_{UL}} \right\rceil + Pr(6(B)) + T(BAck) + 2 \cdot SIFS$$

$$T(BAck) = TSym_{leg} \cdot \left\lceil \frac{(54 \cdot 8) + 22}{TSym_{leg} \cdot R_{leg}} \right\rceil$$

$$T(CF - End) = TSym_{leg} \cdot \left\lceil \frac{(20 \cdot 8) + 22}{TSym_{leg} \cdot R_{leg}} \right\rceil \quad (2)$$

$T(BAck)$ and $T(CF - End)$ are the transmission times of the BAcK and the CF-End frames respectively. The times are based on the frames' lengths shown in Figure 5. $T(BAck)$ assumes the BAcK frame acknowledging 256 MPDUs, Figure 5(B). If the number of MPDUs is smaller than or equal to 64, the BAcK frame of Figure 5(A) is used, and in this case the term 54 in the numerator of $T(BAck)$ is replaced by 30.

$TSym_{DL}$ and $TSym_{UL}$ are the lengths of the OFDM symbols in the DL and UL respectively, including the GI, when transmitting in a non-legacy mode, and every transmission must be of an integral number of OFDM symbols. $TSym_{leg} = 4 \mu s$ and it is the length of the OFDM symbol that is used in legacy transmissions. R_{DL} and R_{UL} are the PHY rates for TCP Data/Ack transmissions on the DL and the

UL respectively, and R_{leg} is the legacy PHY rate used for the transmission of the BAck and CF-End frames. See Table 1. The additional 22 bits in the various numerators of the frames' transmission times are due to the SERVICE and TAIL fields that are added to every transmission by the PHY layer conv. protocol [1].

4.1.2 Goodput analysis for Scheduling strategy 2 - HE DL Single User contention based unidirectional TCP

In this scheduling strategy the AP transmits N DL TCP Data segments, $1 \leq N \leq N_{MAX}$, to a station after accessing the channel. The number of HE DL SU A-MPDU frames containing these N TCP Data segments and their structure is determined as in scheduling strategy 1. The AP does not transmit to a station again before it receives UL TCP Acks from the station and transmits to the stations in Round Robin fashion. We check the performance of this scheduling strategy for various values of N , the number of DL TCP Data segments that the AP transmits in one transmission to the station, $1 \leq N \leq N_{MAX}$.

4.1.3 Goodput analysis for Scheduling strategy 3 - HE DL simultaneous Multi User unidirectional TCP

In scheduling strategy 3 every single DL TCP connection between the AP and a station can be considered as the TCP connection in scheduling strategy 1. The only difference is that in scheduling strategy 3 the AP can transmit in several DL TCP connections parallel to several stations, and several stations can transmit their BAck and UL TCP Ack segments parallel to the AP. The analysis in scheduling strategy 3 is therefore basically the same as for scheduling strategy 1 with some differences specified below.

The Goodput of scheduling strategy 3 shown in Figure 4 is given by Eq. 3, assuming the AP transmits N DL TCP Data MSDUs in every TCP connection in n HE DL MU A-MPDUs:

$$Goodput = \frac{N \cdot (L_{DATA} \cdot 8) \cdot S}{AIFS + BO(Variable) + \sum_{i=1}^n T_{Data}^{cycle}(A_i, N_i) + T_{Ack}^{cycle}} \quad (3)$$

$$T_{Data}^{cycle}(A_i, N_i) = Pr(6(C)) + TSym_{DL} \cdot \left[\frac{(N_i \cdot Len^D + A_i \cdot (OM + 4)) \cdot 8 + 22}{TSym_{DL} \cdot R_{DL}} \right] + Pr(6(D)) + T(BAck) + 2 \cdot SIFS$$

$$T_{Ack}^{cycle} = Pr(6(B)) + T(TF) + Pr(6(D)) + TSym_{UL} \cdot \left[\frac{(N \cdot Len^A + \left\lceil \frac{N}{178} \right\rceil \cdot OM) \cdot 8 + 22}{TSym_{UL} \cdot R_{UL}} \right] + Pr(6(B)) + T(Mul.BAck) + 2 \cdot SIFS$$

$$T(BAck) = TSym_{UL} \cdot \left[\frac{(54 \cdot 8) + 22}{TSym_{UL} \cdot R_{UL}} \right]$$

$$T(TF) = TSym_{leg} \cdot \left[\frac{(28 + (\frac{S}{2} \cdot 5) \cdot 8) + 22}{TSym_{leg} \cdot R_{leg}} \right]$$

$$T(Mul.BAck) = TSym_{leg} \cdot \left[\frac{((22 + S \cdot 36) \cdot 8) + 22}{TSym_{leg} \cdot R_{leg}} \right] \quad (4)$$

The quantity 4 in the numerator of the second term in $T_{Data}^{cycle}(A_i, N_i)$ stands for the HE IE added to every MPDU in order to schedule parallel transmissions of the BAck frames from the stations on the

UL. This holds for $1 \leq A_i \leq 18$. For $A_i > 18$ it is more efficient to contain a unicast TF frame of length 72 bytes, containing the unicast TF frame (33 bytes) and the MAC Header, MPDU Delimiter and FCS fields, and rounding to an integral multiple of 4 bytes.

$T(BAck)$, $T(TF)$ and $T(Mul.BAck)$ are the transmission times of the BACk, multicast TF and the Multi Station BACk frames respectively. $T(MUl.BAck)$ is based on the Multi Station BACk frame length given in Figure 5(C) assuming the acknowledgment of 256 MPDUs per HE DL A-MPDU frame. When considering the acknowledgment of 64 MPDUs the term 36 in the numerator is replaced by 12. The term S in $T(TF)$ and $T(Mul.BAck)$ denotes the number S of stations transmitting data simultaneously over the UL.

$TSym_{UL}$ and $TSym_{DL}$ are the lengths of the OFDM symbols, containing the GIs, used over the UL and DL respectively when transmitting in non-legacy mode, and every transmission must be of an integral number of OFDM symbols. $TSym_{leg} = 4\mu s$ and it is the length of the OFDM symbol that is used in legacy transmissions. R_{DL} and R_{UL} are the DL and UL PHY rates respectively used for the transmission of TCP Data and TCP Acks segments. R_{leg} is the legacy PHY rate used for the transmission of the TF and Multi Station BACk frames. The additional 22 bits in the various numerators of the frames' transmission times are due to the SERVICE and TAIL fields that are added to every transmission by the PHY layer conv. protocol [1].

4.2 Goodput vs. Delay computation

For scheduling strategies 1 and 3 we measured the Goodput received in every TXOP according to Eqs. 2 and 4 respectively. In these equations the total number of TCP Data bits transmitted in a TXOP is divided by the TXOP length, measured in seconds. However, since we assume that the same TXOPs repeat themselves one after another, the computed Goodput of a TXOP is also the Goodput of the system.

We now measure for every number N of TCP Data segments transmitted in a TXOP, $1 \leq N \leq N_{MAX}$ the resulted length of the TXOP interval containing the N TCP Data segments' and as mentioned the Goodput is computed using Eqs. 2 and 4 respectively.

For scheduling strategy 2 we also measure the Goodput when transmitting N TCP Data segments. However, in this scheduling strategy there is no TXOP with RD and instead we measure the average time elapsed from the time the AP transmits to a station TCP Data segments until it receives TCP Acks from the station.

From now on we denote by *cycle* the TXOPs in scheduling strategies 1 and 3, and the above time interval that we described for scheduling strategy 2. By *cycle length* we denote the length, in seconds, of the cycle.

The next step is as follows: Notice that for every number N of TCP Data segments transmitted in a cycle, there is a resulting cycle length which shows how much time is needed in a cycle for the transmission of these TCP Data segments to a specific station.

Thus, for every number N of TCP Data segments, $1 \leq N \leq N_{MAX}$ we attach two measures:

the cycle length in which these N TCP Data segments are transmitted and the resulting Goodput. We now arrange the cycle lengths in a list together with the associated Goodputs in increasing order of the cycle lengths.

Notice that two different cycle lengths can have the same Goodput. One of the cycles has more TCP Data segments but it can also have more A-MPDU/MPDUs' overhead. In addition, the number of TCP Data segments can be large enough so that the addition of one more TCP Data segment barely changes the Goodput. For a set of cycle lengths with the same Goodput we leave only the shortest cycle in the list.

Consider now a cycle length of L ms with a Goodput G . In scheduling strategies 1 and 2 (the SU ones) when the AP is communicating with S stations in Round Robin, a station receives TCP Data segments in every S^{th} cycle. Thus, a station receives a service for L ms with a Goodput G , and then waits $(S-1) \cdot L$ ms before receiving TCP Data segments again. In total the system provides a Goodput G for all stations during an interval of $S \cdot L$ ms. In scheduling strategy 3 (the MU one) where S stations transmit in a TXOP, every station has a Goodput $\frac{G}{S}$ during an interval of L ms. Overall the system provides a Goodput G to all the stations during every interval of L ms.

5 Goodput results

In Figure 7 we plot 6 graphs showing the Goodput of the system vs. the delay (cycle length) for the cases of $S = 1, 4, 8, 16, 32$ and 64 stations in Figures 7(A), (B), (C), (D), (E) and (F) respectively, for TCP Data segments of length 1460 bytes and for the case where Delayed Acks are not used, i.e. every TCP Ack acknowledges one TCP Data segment. The results for TCP Data segments of 464 and 208 bytes are similar. As mentioned, the graphs for scheduling strategies 1 and 3 were obtained by analysis and simulation. The results for scheduling strategy 2 were obtained by simulation only.

In Figure 7(A) we show results for a single station and therefore only scheduling strategies 1 and 2 are relevant. We show results for MCSs 1, 3, 5, 7, 9 and 11. We see that the two scheduling strategies have similar results because there are no collisions in scheduling strategy 2 - the AP and the single station transmit alternately.

In Figure 7(B) we show results for 4 stations and for MCSs 5 and 11, this time for all the scheduling strategies. The results for all the other MCSs are similar. We see that the maximum Goodput is received in MU for about 300ms while in SU the same maximum Goodput is received in much longer times, i.e. more TCP Data segments need to be transmitted. Therefore, the MU strategy outperforms the SU strategies, while using RD outperforms the contention based strategy. We can therefore conclude that the MU uses the channel more efficiently in this case, and enables a better performance for TCP than SU.

The same result also holds for 8 stations, Figure 7(C). In the case of 16 stations the MU strategy almost achieves the maximum Goodput. The RD strategy achieves the maximum Goodput, although in much larger delays. The MU strategy has small PHY rates that do not enable transmission of many TCP Acks due to the limit on transmission time of the HE UL MU A-MPDU frame containing the TCP Acks. As a consequence the number of TCP Data segments that can be transmitted in a TXOP is

relatively small. Therefore, it is not possible to transmit as many TCP Data segments in a TXOP as in the SU strategies, and the resulting Goodput is smaller.

Notice that the above phenomena is also observed in the case of 32 stations, Figure 7(E). In Figure 7(F), the case for 64 stations, the very small PHY rates in MU cause the SU modes to outperform MU significantly.

In Figure 8 we show the same results as in Figure 7, this time with results for the Delayed Acks feature. For 4, 8, 16, 32 and 64 stations we show results only for MCS11. It can be concluded that the improvement in the Goodput is only marginal. Using Delayed Acks enables transmission of more TCP Data segments in a cycle. However, there is no save in the overhead of A-MPDUs and MPDUs containing TCP Data segments. There is only a small save in the overhead involved in the transmission of the TCP Acks, which is marginal, and so is the Goodput gain.

In Figure 9 we show results for the various TCP Data segments' sizes, 208, 464 and 1460 bytes for MCS11, for the cases of 4, 8 and 16 stations in Figures 8 (A), (B) and (C) respectively. We also show results with and without Delayed Acks. Since the number of TCP Acks that can be transmitted in a cycle does not change, one can expect that as the length of the TCP Data segments decreases, the length of the respective cycles also decrease. This also is true for the respective Goodputs since the overhead of transmitting TCP Ack segments remains unchanged.

We see these expected results in Figure 9. Notice that for all cases the curves end at the longest cycles possible and these lengths decrease as the TCP Data segment lengths decrease.

We can also see that while for TCP Data segments of 1460 bytes the use of Delayed Acks results only in marginal Goodput improvement, the other TCP Data segments' lengths such as 464 and 208 bytes show significant improvement. in the order of 15 – 20%. With short TCP Data segments one can add many such segments without increasing the number of MPDUs and A-MPDUs significantly, while greatly increasing the number of TCP Data bytes transmitted. Therefore, the ratio between the increase in the TCP data to the increase in the A-MPDUs/MPDUs overheads is much better than in the case of long TCP Data segments and the increase in the Goodput is more significant.

6 Summary

In this paper we have introduced three scheduling strategies for the transmission of TCP Data over the DL of an IEEE 802.11ax system, where the AP is the TCP Data transmitter and the stations are the receivers. Two of the strategies are SU, and one strategy is a MU. We measured the Goodput of the system as a function of the time it takes the system to provide this Goodput.

We found that for up to 8 stations the MU strategy outperforms those of SU, i.e. the maximum Goodput is achieved in the MU strategy in much shorter time intervals than in

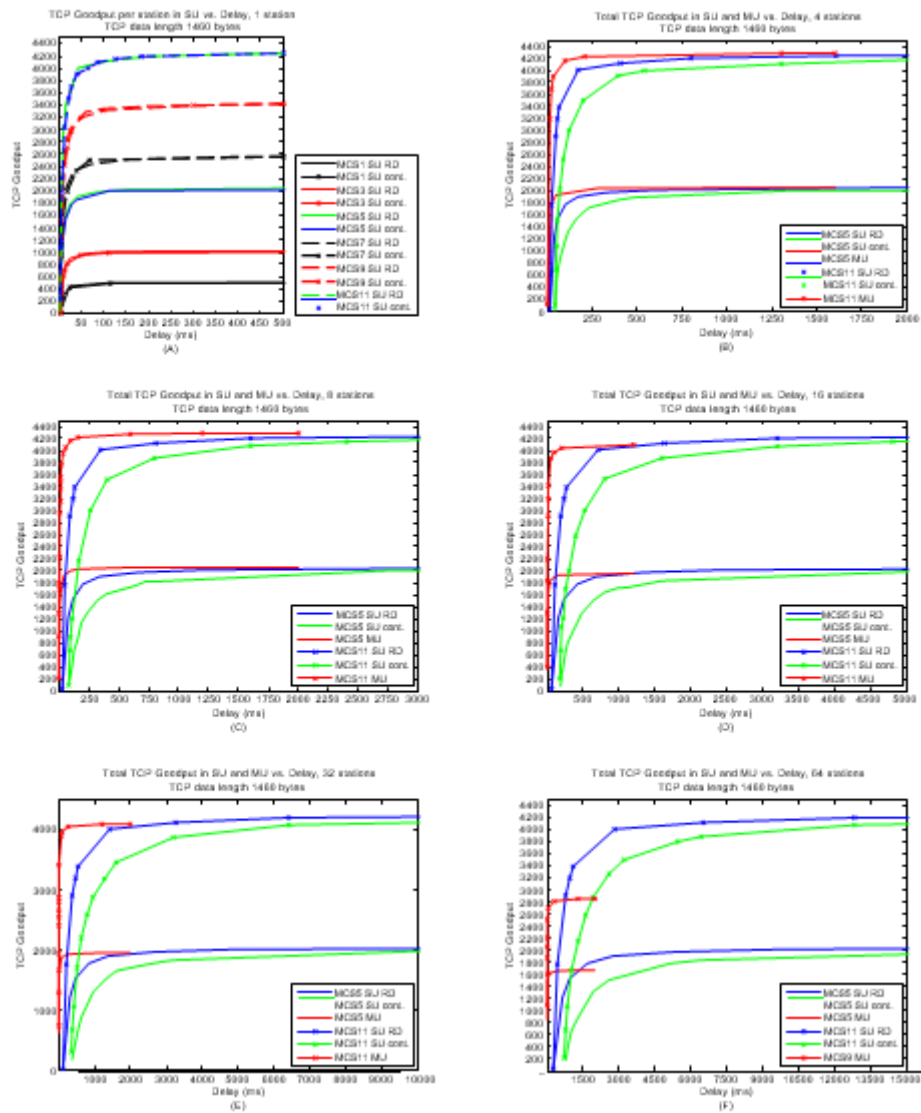


Figure 7: Total TCP Goodput vs. delay in Scheduling strategies 1, 2 and 3 in MCS 5 and MCS 11. 1, 4, 8, 16, 32 and 64 stations. TCP data length 1460 bytes.

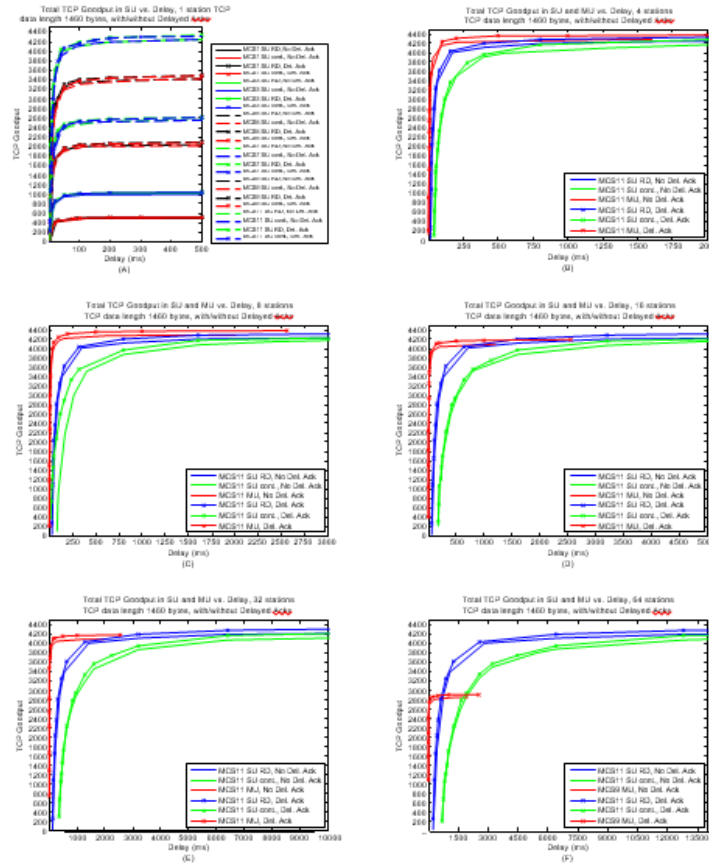


Figure 8: Total TCP Goodput vs. delay in Scheduling strategies 1, 2 and 3 in MCS 11. 4, 8, 16, 32 and 64 stations. TCP data length 1460 bytes. Delayed Acks vs. No Delayed Ack.

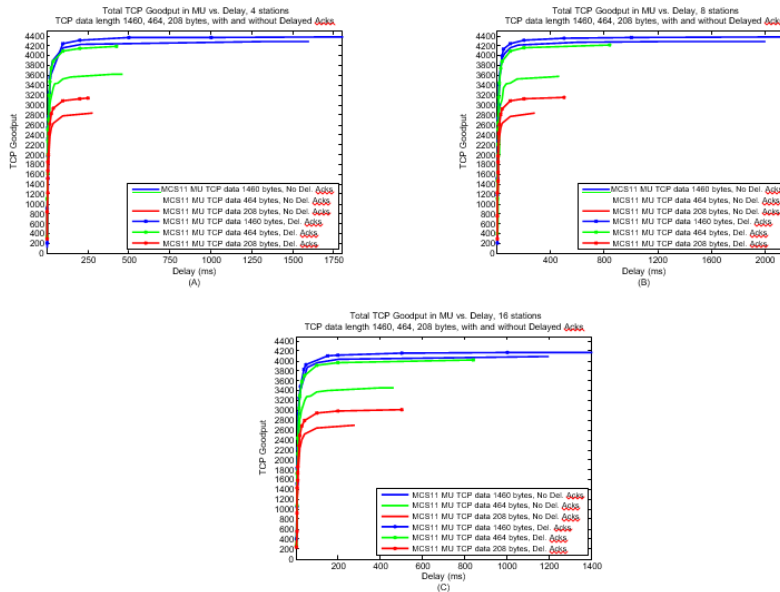


Figure 9: Total TCP Goodput vs. delay in MU - scheduling strategy 3 . Number of stations: 4, 8 and 16. TCP data lengths 1460, 464 and 208 bytes.

the SU strategies. For the case of 16 and 32 stations the MU strategy achieves almost the same Goodput of the SU strategies, but does so in much shorter time intervals. The SU strategies achieve a slightly larger Goodput but with much longer time intervals. Therefore, in these cases the SU and MU strategies have about the same performance. For the case of 64 stations the SU strategies are much better than the MU because the latter has very small PHY rate channels.

Finally, we found that using Delayed Acks has only marginal influence on the Goodput when transmitting long TCP Data segments. The Delayed Acks feature results with significant improvement in the achieved Goodput, in the order of 15 – 20%, when the TCP Data segments are short.

7 Appendix

In this appendix we show how to schedule N TCP Data MSDUs into MPDUs and A-MPDUs frames in the HE DL TCP Data cycles of various scheduling strategies, in a way that minimizes the MPDUs' overhead when the number of A-MPDUs is given. We first compute upper and lower limits on the number of A-MPDUs needed for the transmission of N TCP Data MSDUs. For every number of A-MPDUs in the range we show scheduling that uses a minimal number of MPDUs, i.e. the smallest MPDUs' overhead.

We begin with several definitions.

Definitions:

1. A_{Data} : The maximum possible number of TCP Data MSDUs in an MPDU.
2. **Full MPDU**: An MPDU containing A_{Data} TCP Data MSDUs.
3. **Partial MPDU**: An MPDU containing less than A_{Data} TCP Data MSDUs.
4. F_{Data} : The maximum possible number of Full MPDUs in an A-MPDU.
5. **Full A-MPDU**: An A-MPDU containing the maximum possible number of TCP Data MSDUs.
6. **Partial A-MPDU**: An A-MPDU that contains less than the maximum possible number of TCP Data MSDUs.

Consider an A-MPDU frame that contains F_{DATA} Full MPDUs and possibly one more Partial MPDU. The Partial MPDU contains the maximum possible number of MSDUs given the limit on transmission time of a PPDU. Notice that this limit is the reason why it is not always the case that an A-MPDU contains only Full MPDUs. We denote such a scheduling of MSDUs within an A-MPDU by F -construction.

Claim 1: An A-MPDU with an F -construction is a Full A-MPDU.

Proof: Let X and Y be the numbers of the MPDUs and MSDUs in the F -construction respectively. Assume it is possible to schedule $Y' > Y$ MSDUs in an A-MPDU given the limit on the transmission time of a PPDU.

It must hold that the Y' MSDUs are scheduled in the A-MPDU within X or more MPDUs because it is immediately seen that X is the minimum number of MPDUs necessary for the transmission of Y MSDUs.

It is not possible to schedule the Y' MSDUs in X MPDUs only because this will violate the limit on the transmission time of the PPDU containing the A-MPDU. Therefore, at least $X+1$ MPDUs are required. This means that the transmission time of the A-MPDU with Y' MSDUs is larger than that of an A-MPDU with an F -construction by at least the transmission time of one MPDUs' overhead and one MSDU. However, this violates the given that it is not possible to add even one more MSDU to an A-MPDU with an F -construction without violating the transmission time of a PPDU. ■

Let $X_{min} = \left\lceil \frac{N}{F_{DATA}} \right\rceil$ be the minimum number of MPDUs necessary to contain N TCP Data MSDUs. It is possible to schedule the MSDUs into the MPDUs such that $\left\lceil \frac{N}{F_{DATA}} \right\rceil - 1$ MPDUs are Full MPDUs and one more MPDU is either a Full or a Partial MPDU.

Let $X_{upper} = \left\lceil \frac{X_{min}}{F_{DATA}} \right\rceil$ be an upper limit on the number of A-MPDUs that are needed to contain the X_{min} MPDUs. $(X_{upper} - 1)$ A-MPDUs contain F_{DATA} MPDUs and the last A-MPDU possibility contains less than F_{DATA} MPDUs and possibly one Partial MPDU.

Let X_{lower} be a lower limit on the number of A-MPDUs needed to transmit the N TCP Data MSDUs. One possibility for scheduling the N TCP Data MSDUs in these X_{lower} A-MPDUs is by defining $(X_{lower} - 1)$ Full A-MPDUs and possibly one Partial A-MPDU.

Notice that by using X_{upper} A-MPDUs one uses the smallest amount of overhead caused by MPDUs and the largest amount of overhead caused by A-MPDUs. The MPDU's overhead is the MAC Header, MPDU Delimiter and the FCS fields. The A-MPDU overhead for scheduling strategies 1 and 2 is $Pr(B(A)) + Pr(B(B)) + T(BAck) + 2 \cdot SIFS$. For scheduling

strategy 3 the presumes are $Pr(\mathcal{B}(C))$ and $Pr(\mathcal{B}(D))$.

By using X_{lower} A-MPDUs one uses the largest amount of overhead caused by MPDUs and the smallest amount of overhead caused by A-MPDUs. To find the maximum Goodput when transmitting N TCP Data MPDUs, one needs to review all numbers of A-MPDUs X , $X_{lower} \leq X \leq X_{upper}$, and determine the minimum MPDUs' overhead when using X A-MPDUs. We then need to find the minimum sum of overheads of both A-MPDUs and MPDUs for all X s in the range.

In the following we show the scheduling that results with the smallest amount of MPDUs' overhead given N TCP Data MSDUs and X A-MPDUs,

For this purpose we now define the following scheduling α of N TCP Data MSDUs into X A-MPDUs, $X_{lower} \leq X \leq X_{upper}$.

Scheduling α :

- $X = X_{upper}$: $(X_{upper} - 1)$ A-MPDUs contain only F_{DATA} Full MPDUs. The last A-MPDU contains as many as possible Full MPDUs and possibly one Partial MPDU.
- $X_{lower} \leq X \leq (X_{upper} - 1)$: All the A-MPDUs contain F_{DATA} Full MPDUs. The remaining MSDUs are used to construct as many as possible Full A-MPDUs (F-construct).

Claim 2: Assume X A-MPDUs, $X_{lower} \leq X \leq (X_{upper} - 1)$ in scheduling α , such that every A-MPDU contains F_{DATA} Full MPDUs. Scheduling α then contains the minimal number of Partial MPDUs needed for the scheduling of the remaining $M = N - A_{data} \cdot F_{DATA} \cdot X$ TCP Data MSDUs in the X A-MPDUs.

Proof: Assume that every Partial MPDU in an A-MPDU with F_{DATA} Full MPDUs can contain at most M' MSDUs. To schedule M MSDUs in such Partial MPDUs one needs at least $\left\lceil \frac{M}{M'} \right\rceil$ such MPDUs, which is the number of MPDUs that scheduling α uses. ■

Claim 3: Let P be the number of MPDUs defined by scheduling α when scheduling N TCP Data MSDUs in X A-MPDUs, $X_{lower} \leq X \leq (X_{upper} - 1)$. P is then the minimal number of MPDUs necessary to schedule N TCP Data MSDUs in the X A-MPDUs.

Proof: Assume that $N - A_{DATA} \cdot F_{DATA} \cdot X + M, M \geq 0$ and thus M is the number of MSDUs in Partial MPDUs in scheduling α .

Assume on the contrary that there is another scheduling β in which the N TCP Data MSDUs are scheduled in X A-MPDUs within less than P MPDUs.

Notice that if an A-MPDU in scheduling β contains two or more Partial MPDUs then it is possible to re-arrange the scheduling of the MSDUs within the A-MPDU such that the number of MPDUs is not changed and that all the MPDUs in the A-MPDU are Full MPDUs except possibly one Partial MPDU.

We re-arrange all the A-MPDUs in scheduling β as described above. We go then through the A-MPDUs and for every A-MPDU number t , $1 \leq t \leq X$, we do the following: If A-MPDU t does not contain F_{DATA} Full MPDUs, we borrow MSDUs from Partial MPDUs in other A-MPDUs and schedule them in A-MPDU t such that we form F_{DATA} Full MPDUs in A-MPDU t . Notice that such a process does not increase the overall number of MPDUs; if a new MPDU is generated in A-MPDU t , a Partial MPDU in another A-MPDU must be canceled.

The above process can be performed over all the A-MPDUs as there are at least $A_{DATA} \cdot F_{DATA} \cdot X$ MSDUs to schedule. During the process the number of MPDUs does not increase, but does not decrease due to the optimality of scheduling β . By Claim 2 scheduling β cannot have a smaller number of MPDUs than scheduling α . ■

Lemma 1: For X A-MPDUs, $X_{lower} \leq X \leq X_{upper}$ and N TCP Data MSDUs, scheduling α results in the smallest MPDUs' overhead.

Proof: For every X in the given range, scheduling α generates the smallest possible number of MPDUs needed for the scheduling of N TCP Data MSDUs. Therefore, scheduling α results in the smallest MPDUs' overhead. ■

REFERENCES

- [1] IEEE Std. 802.11T M -2016, IEEE Standard for Information Technology - Telecommu- nications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, NewYork, (December 2016)
- [2] IEEE P802.11axT M /D1.4, IEEE Draft Standard for Information Technology - Telecom- munications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specific requirements. IEEE, NewYork, (2017)

- [3] IEEE Std. 802.11acT M -2013, IEEE Standard for Information Technology - Telecommu- nications and Information Exchange between Systems - Local and Metropolitan Area Networks - Specific Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specific requirements. Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz, IEEE, NewYork, (2013)
- [4] E. Perahia, R. Stacey, Next Generation Wireless LANs: 802.11n and 802.11ac, 2nd Edition, Cambridge Press, 2013
- [5] E. Khorov, A. Kiryanov, A. Lyakhov, IEEE 802.11ax: How to Build High Efficiency WLANs, Int. Conf. on Eng. and Telecommunication (2015) 14-19
- [6] M. S. Afaqui, E. G. Villegas, E. L. Aguilera, IEEE 802.11ax: Challenges and Re- quirements for Future High Efficiency WiFi, IEEE Wireless Communications 99 (2016) 2-9
- [7] D. J. Deng, K. C. Chen, R. S. Cheng, IEEE 802.11ax: Next Generation Wireless Local Area Networks, 10th Int. Conf. on Heterogeneous Networking for Quality, Security and Robustness (QSHINE), (2014) 77-82
- [8] B. Bellalta, IEEE 802.11ax: High-efficiency WLANs, IEEE Wireless Communications, 23(1) (2016) 38-46
- [9] O. Sharon, Y. Alpert, Scheduling strategies and Throughput optimization for the Uplink for IEEE 802.11ax and IEEE 802.11ac based networks, Wireless Sensor Networks,9 (2017) pp. 250-273
- [10] Sharon, Y. Alpert , Scheduling strategies and throughput optimization for the Downlink for IEEE 802.11ax and IEEE 802.11ac based networks, Wireless Sensor Networks, 9 (2017) pp. 355-383
- [11] O. Sharon, Y. Alpert, Single User MAC level Throughput comparision: IEEE 802.11ax vs. IEEE 802.11ac, Wireless Sensor Networks, 9 (2017), pp. 166-177
- [12] R. Karmakar, S. Chattopadhyay, S. Chakraborty, Impact of IEEE 802.11n/ac PHY/MAC High Throughput Enhancement over Transport/Application layer proto- cols - A Survey, IEEE Communication surveys and tutorials (2017)
- [13] Miorandi, D., Kherani, A. A. and Altman, E. (2006) A Queueing model for HTTP traffic over IEEE 802.11 WLANs. Computer Networks, 50, 63-79.
- [14] Bruno, R., Conti, M. and Gregori, E. (2005) Throughput Analysis of UDP and TCP Flows in IEEE 802.11b WLANs: A Simple Model and its Validation. Workshop on Techniques, Methodologies and Tools for Performance Evaluation of Complex Systems, 2005, 54-63.
- [15] Bruno, R., Conti, M. and Gregori, E. (2008) Throughput Analysis and Measurements in IEEE 802.11 WLANs with TCP and UDP Traffic Flows. IEEE Trans. on Mobile Computing, 7, 171-186.

- [16] Kumar, A., Altman, E., Miorandi, D. and Goyal, M. (2007) New Insights from a Fixed Point Analysis of Single Cell IEEE 802.11 WLANs. *IEEE/ACM Trans. on Networking*, 15, 588-601.
- [17] Q. Qu, B. Li, M. Yang, Z. Yan, An OFDMA based Concurrent Multiuser MAC for Upcoming IEEE 802.11ax, *IEEE Wireless Comm. and Networking Conf. Workshops (WCNCW) (2015)* 136-141
- [18] W. Lin, B. Li, M. Yang, Q. Qn, Z. Yan, X. Zuo, B. Yang, Integrated Link-System level Simulation Platform for the Next Generation WLAN - IEEE 802.11ax, *IEEE Globecom (2016)*
- [19] J. Lee, D. J. Deng, K. C. Chen, OFDMA-based hybrid channel access for IEEE 802.11ax WLAN, unpublished.
- [20] M. Karaca, S. Bastani, B. E. Priyanto, M. Safavi, B. Landfeldt, Resource Management for OFDMA based Next Generation 802.11ax WLANs, *9th IFIP Wireless and Mobile Networking Conf. (WMNC) (2016)*
- [21] V. Jones, H. Sampath, Emerging technologies for WLAN, *IEEE Commu. Mag.*, 5 (2015) 141-9
- [22] L. Sanabria-Russo, A. Faridi, B. Bellalta, J. Barcelo, M. Oliver, Future evolution of CSMA protocols for the IEEE 802.11 standard *IEEE Int. Conf. on Comm. (ICC) (2013)* 1274-9
- [23] L. Sanabria-Russo, J. Barcelo, A. Faridi, B. Bellalta, WLANs throughput improvement with CSMA/ECA, *IEEE Conf. on Computer Comm. Workshops (INFOCOM WKSHPS) (2014)* 125-6
- [24] Y. He, R. Yuan, J. Sun, W. Gong, Semi-Random Backoff: Towards resource reservation for channel access in wireless LANs, *IEEE Int. Conf. on Network Protocols (ICNP) (2009)* 21-30
- [25] E. Khorov, V. Loginov, A. Lyakhov, Several EDCA Parameters Sets for Improving Channel Access in IEEE 802.11ax Networks, *Int. Symposium on Wireless Communication Systems (ISWCS) (2016)* 419-423
- [26] O. Sharon, Y. Alpert, Coupled IEEE 802.11ac and TCP performance evaluation in various aggregation schemes and Access Categories, *Computer Networks* 100 (2016) 141-156
- [27] O. Sharon, Y. Alpert, Couples IEEE 802.11ac and TCP Goodput improvement using Aggregation and Reverse Direction, *Wireless Sensor networks* 8(7) (2016) 107-136
- [28] O. Sharon, Y. Alpert, Comparison between TCP scheduling strategies in IEEE 802.11ac based Wireless networks, *Ad Hoc Networks* 61C (2017) pp. 95-113
- [29] O. Sharon, Y. Alpert, MAC level Throughput comparison: 802.11ac vs. 802.11n, *Physical Communication Journal* 12 (2014) 33-49
- [30] IEEE Std. 802.11T M -2012, Standard for Information Technology - Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks Specific

Requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, IEEE, NewYork, (2012)

- [31] Network Simulator 2 (NS-2). Available: <http://www.isi.edu/nsnam/ns>.
- [32] E. Modiano, An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols, *Wirel. Netw.* 5 (1999) 279-286.
- [33] F. Yang, Q. Zhang, W. Zhu, Y.Q. Zhang, An efficient transport scheme for multimedia over wireless internet, *IEEE Int. Conf. on 3G wireless and beyond (3Gwireless)*, 2001
- [34] K. Ramin, K. Sabmatian, Evaluation of packet error rate in wireless networks, *7th IEEE/ACM Int. Sym. on Mobility, Analysis and Simulation of Wireless Mobile Systems (MSWIM)*, 2004
- [35] D. Gomez, R. Agüero, M. Garcia-Arranz, L. Muñoz, Replication of the Bursty Behavior of Indoor WLAN Channels, *Proc. of the 6th Int. ICST Conf. on Simulation Tools and Techniques*, (2013) pp. 219-226

Multiplexing and De Multiplexing using Visual Basic Program

Ahmad A. Mohammed AL-Mafri

Department of mathematics, College of Education for pure Sciences, University of Kirkuk, Iraq.
ahabm2012@yahoo.com

ABSTRACT

The strategy of the global telecommunications companies always provide the best services to users is the high speed in the transfer of reliable data and a large amount of data (calls) with the least effort and cost. This study focuses on the use of simulations for the implementation of Multiplexing technique and DE Multiplexing technique through programming using Visual Basic language. The simulation helps to give an easy and clear idea of how Multiplexing technique and DE Multiplexing technique work. The system was created and implemented using the Visual Basic 8.0.

Keywords:- Multiplexer technique, De Multiplexer technique, Visual Basic language.

1 Introduction

1.1 Multiplexer

Multiplexer circuits find many different implementations in digital systems of various kinds. These implementations include data selection, data routing, process sequence, parallel conversion to serial, waveform generation, generation logic function.[1]. Multiplexer works like a multi-digit digitally controlled switch where a numeric code is applied to selective input control that input data will be converted to an output. This is called multiplexing.[2]. The Multiplexer is a synthesized circuit which identifies binary data from one of the many input lines and directs it to a one output line .The selection of a specific input line is controlled by the set of selection lines.[3]. Typically, there are 2^n input lines and n lines of selection which bit sets define which inputs are selected .

1.2 Demultiplexer

A Demultiplexer is a logical circle with an input line, and $2n$ and n output lines specify lines[4]. It directs the information on the input line to any of the production lines [5]. The output line that obtains information on the input line is determined by the bit status of the selection lines [6]

1.3 Visual Basic Language

Visual Basic first appeared in 1991, and since then changes have been made to this language with the emergence of different versions. Visual Basic is a Windows programming language. It is used to design programs running under Windows.

Therefore, those who want to learn this language must be familiar with the Windows operating system, preferably familiar with Basic programming language In Visual Basic, the similarities are very similar to those in Basic [7]. Visual Basic is an event- oriented language, as are most Windows programming languages (Delphi and Visual C ++). Event-driven language is the language in which the idea of partitioning the program is based on partial programs execute when an event occurs, such as pressing the button or moving a mouse pointer over a window or the passage of time. Therefore, the program have to determine the events and how to respond to them [8].

2 Related Work

The authors in [9] presented a new technique that simulation process of interfering with the package using the OPTI system. The multiplexing process of multiplexing is an extension of the technique known as multiplexing by dividing the electrical time in the optical field. Where optical data streams are created by multiplexing the transmission time by the number of optical streams at a lower bit rate.

The authors in [10] suggested an approach for designing four users Optical Time Division Multiplexing OTDM on single mode fiber (SMF) for different bit rates have been successfully evaluated. The performance of the system depends on both the bit rate and the length of the fiber. The use of EDFAs enhances system performance and link length. The simulation was implemented using Optisystem 7 and Matlab R2011a.

The authors in [11] The strategy provided by Optical Time Division Multiplexing (OTDM) is to increase the bit rate of the digital fiber opticl system over the bandwidth capacity of the electronics engine. Different fiber parameters and different shaping patterns were used with the determination of transmission performance after the OTDM interface analysis showing the effect of the inbound signal capability And the signal strength control on the error rate in the bits (BER).

3 Proposed System

Here we present the proposed system for the application of multiplexing technique and De multiplexing using the visual basic programming language. The importance of the program in the implementation of simulation to facilitate and explain how to implement the technology of multiplexing technique and De multiplexing, where we have a computer network of nine computers. Fig.3.1.General definition interface for multiplexing technique and De multiplexing technique.

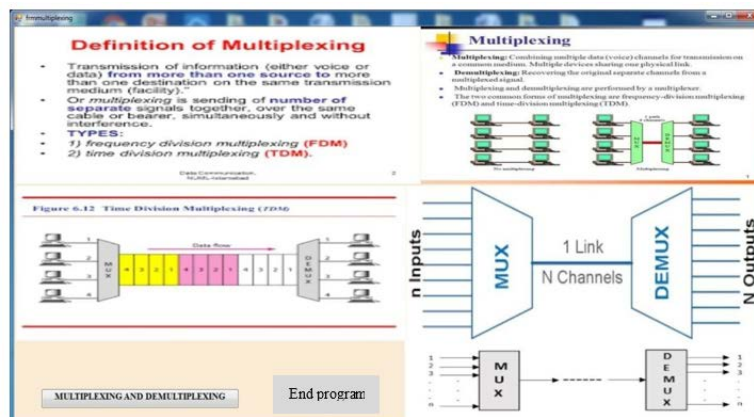


Figure 1 General definition interface for multiplexing technique and De multiplexing technique.

4 Experiments and Results

The set-up parameter for the experiments and the results produced are discussed in detail here. Experiments were conducted to demonstrate the effectiveness of the proposed system. The technology presented in this study is applied to a laptop with the following specifications of the Intel® Core (iM) i5 2430M CPU @ 2.40 GHz 2.40 GHz with 8 GB of RAM, 64-bit system type operating system. This device is equipped with Windows 7 Ultimate operating system. Access to the application program (simulation) after registration of the correct user name and password . And steps to implement the program as following . Fig 1.Login to program

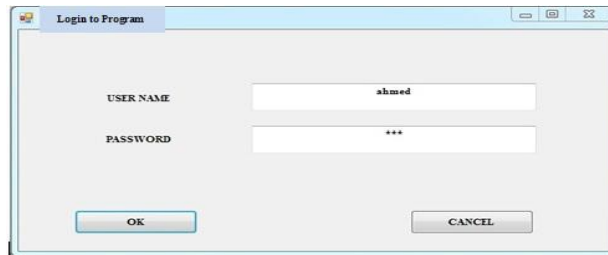


Figure 2 Login to program

1-Write the user name and password correctly for the purpose of entering the program and in the case of wrong entry and three times, the program is closed immediately and is considered a person is not authorized (intruder) as shown in Fig. 2. Attempts login to the program.



Figure 3 Attempts login to the program

2 The Fig. 3. Represents the user name and password correctly .

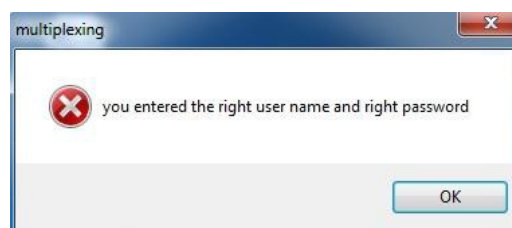


Figure 4 shown user name and password correctly .

3 The Fig.4. Shows the interface of the main application simulation program.

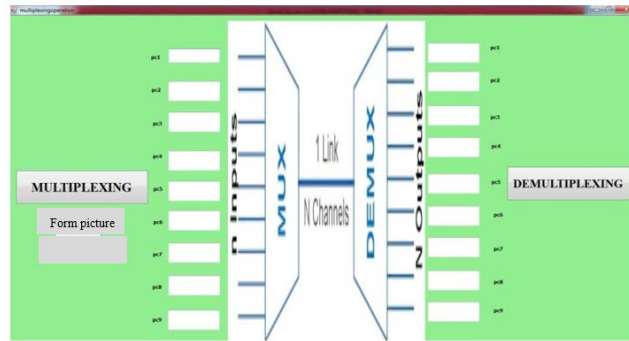


Figure 5 represents the main simulation program interface

4 The Fig. 5. shows the start of data entry in PC1&PC2. Fig. 6. shows the continuous data entry in the computers used in the program from PC1 to PC9.

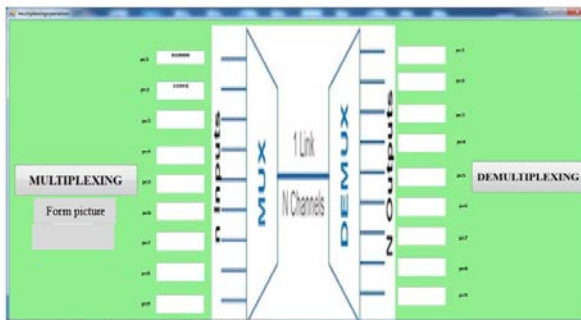


Figure 6 start of data entry in PC 1& PC2

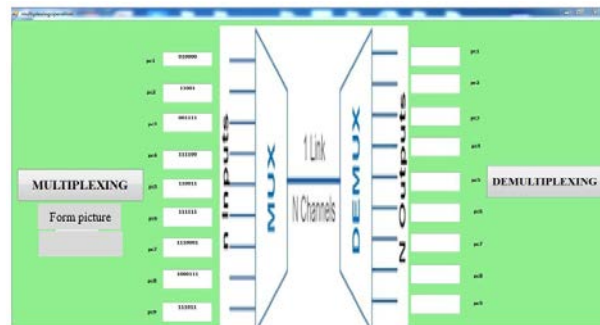


Figure 7 entry the data for all computers

After completing data entry for all computers. Fig. 7. shows start of process loading data in the PC1 to transmission line using multiplexing technique. Fig. 8. shows the complete loading data on all computers from PC1 to PC9 into transmission line using multiplexing technique.

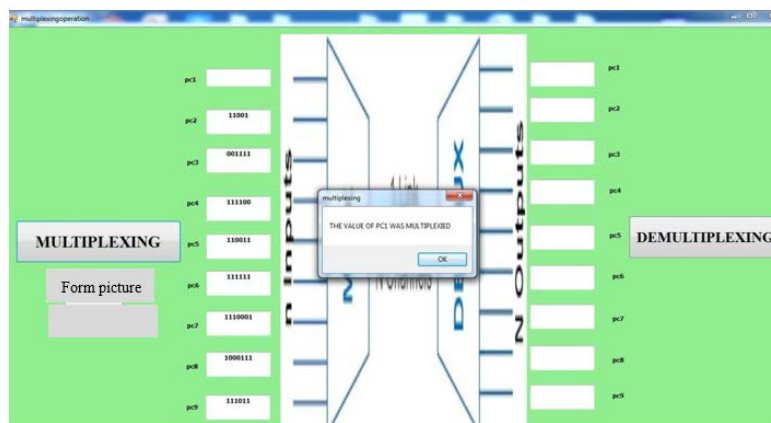


Figure 8 start of process of loading data in the PC1 to transmission line

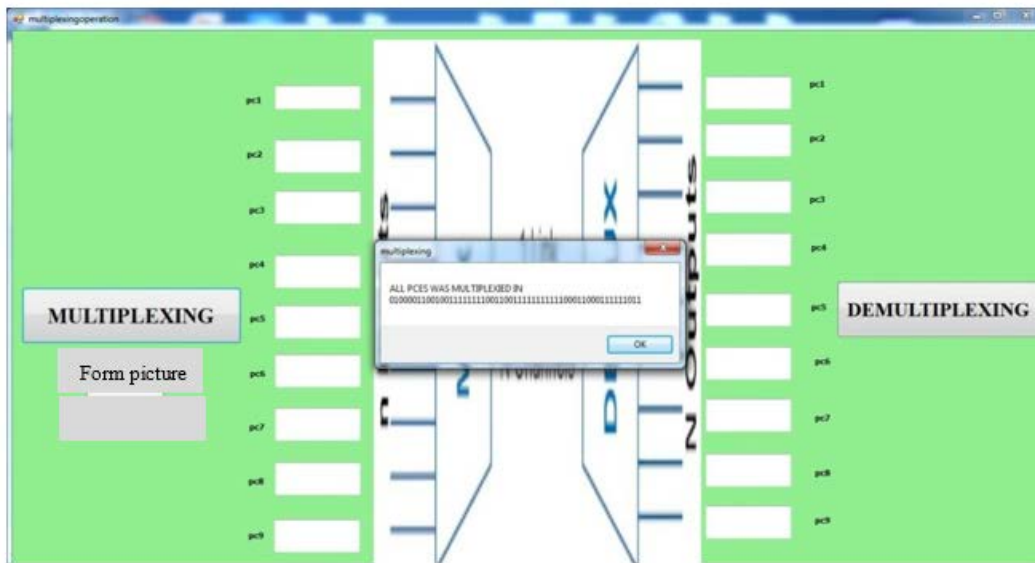


Figure 9 complete loading data for all computers from PC1 to PC9 to transmission line.

5- After completion of the data load to the transmission line (transmitter) begins to use the technique of DE multiplexing where the technique to separate these data from each other and send to the PC required to deliver data to them (PC target)

6- Fig. 9. shows beginning of the transmission data received in the transmission line to PC 1 after separation from each other using a Demultiplexing technique. The distribution of data received via the transmission line to all computers from PC1 to PC9 is completed using Demultiplexing technique as shown in Fig. 10.



Figure 10 start transmission data received to PC1 using Demultiplexing technique.



Figure 11 Data received in transmission line are distributed to all computers

5 Conclusion

The research focuses on use of simulations for purpose of developing multiplexing technique and Demultiplexing technique using Visual Basic language. The results obtained from the experiments concluded that:

- ❖ Using log-in to the program and for a specified number of times gives the importance of maintaining data security.
- ❖ Implementing the program using the simulation becomes clear that it provides better performance and easier understanding of the steps implemented by learners in the area of communication networks.
- ❖ Explain how to load a large amount of data (calls) into a single transmission line (transmission line) using multicast technology.
- ❖ Explain how to separate these data (calls) from each other and send them to the target (s) using the technique of multicast.
- ❖ In conclusion, I say: programming in Visual Basic was really fun language. In very little time we can make good and valuable programs, especially since Visual Basic is a simple to understand matched with other languages such as Visual C ++ and Java.

REFERENCES

- [1] F. O. Moore and E. Osayamwen, "Multiplexing in Networking" Course: MIS7004-2 2017.
- [2] R. J. Tocci, and N. S. Widmer, and G. L. Moss , "Digital Systems Principles and Applications", New Jersey, Pocket Books, 2007.
- [3] M. RAFIQUZZAMAN, "Fundamentals of Digital Logic and Microcomputer Design", United States of America, Pocket Books, 2005.
- [4] M. M. Bruno, and S. Kumar V, "Performance Analysis of OTDM using PON in Direct and External Modulation" IJSR, Vol. 4, Issue 2, February 2015,pp.402-406.

- [5] L. Pasotti, and M. Quattrocelli, and D. Galli, and M. G. De Angelis, and P. Magni, "Multiplexing and demultiplexing logic functions for computing signal processing tasks in synthetic biology" 2013.
- [6] J. Ruela, "Communication Networks", MAP-TELE 2011/12.
- [7] G. Haggrad, and W. Hutchison and Ch. Shibata "Introduction: Visual Basic 6.0" bookboon.com 2013.
- [8] D. I. Schneider, "An Introduction to Programming Using Visual Basic 6.0, Fourth Edition", New Jersey 07458.
- [9] P. N. Kumar, and A. Sangeetha, and G. Srikanth, "analysis of Optical time division multiplexing using packet interleaving scheme" International Journal of Scientific and Research Publications, Vol 3, Issue 4, April 2013, pp.1-5.
- [10] A. Mishra, and P. Mishra, "Optical Communication with Time Division Multiplexing (OTDM) and Hybrid WDM/OTDM PON", International Journal of Science and Research (IJSR), Vol 3 Issue 12, December 2014, pp.1681-1684.
- [11] M. M. Bruno, and S. Kumar V, "Simulation and Performance Evaluation of OTDM in Optical Communication System" IJERT, Vol. 4 Issue 01, January-2015, pp.489 – 493.