Energy-Efficient Bandwidth Allocation for Integrating Fog with Optical Access Networks

by

Ahmed Helmy

A dissertation submitted
in partial fulfillment of the requirements
for the degree of

Doctorate of Philosophy
in Electrical and Computer Engineering

School of Electrical and Computer Science,
Faculty of Engineering,
University of Ottawa,
Ottawa, Ontario

Abstract

Access networks have been going through many reformations to make them adapt to arising traffic trends and become better suited for many new demanding applications. To that end, incorporating fog and edge computing has become a necessity for supporting many emerging applications as well as alleviating network congestions. At the same time, energy-efficiency has become a strong imperative for access networks to reduce both their operating costs and carbon footprint. In this dissertation, we address these two challenges in long-reach optical access networks. We first study the integration of fog and edge computing with optical access networks, which is believed to form a highly capable access network by combining the huge fiber capacity with closer-to-the-edge computing and storage resources. In our study, we examine the offloading performance under different cloudlet placements when the underlying bandwidth allocation is either centralized or decentralized. We combine between analytical modeling and simulation results in order to identify the different factors that affect the offloading performance within each paradigm. To address the energy efficiency requirement, we introduce novel enhancements and modifications to both allocation paradigms that aim to enhance their network performance while conserving energy. We consider this work to be one of the first to explore the integration of fog and edge computing with optical access networks from both bandwidth allocation and energy efficiency perspectives in order to identify which allocation paradigm would be able to meet the requirements of next-generation access networks.

Dedication

To my father, who pushed me to where I am today,

and to my mother, who calls me each and every day..

To my brother, my sister, and their kids whom I adore,

and to Dr. Nour and the family that I'm truly thankful for..

To Shady, my closest friend, who yet lives far away,

and to the one behind my inspiration

more than words could ever say..

Acknowledgements

Writing down this section of my dissertation simply means that the Almighty, praised and glorified be He, has willed for this journey to end and has devised the means for it to conclude as for it to begin. Eternal thanks be to Him, as best as they should, for blessings with no end and for many other things we cannot begin to comprehend.

I would then like to take this opportunity to express my deepest gratitude to my dear supervisor, Prof. Amiya Nayak for his valuable support and continuous assistance. Prof. Amiya accepted to be my supervisor under inconvenient circumstances. He allowed me to think freely and encouraged me to work only in the fields that interest me. His insights and ongoing encouragement have been a constant source of my motivation, especially in targeting and publishing my work in top journals and conferences. My numerous discussions with him have not only increased the quality of my work but have also provided me with valuable advice for my career. I truly enjoyed working with him in both research and teaching.

I must also thank my master advisor, Dr. Habib Fathallah, who always encouraged me to pursue my Ph.D. degree and guided my first steps to become the researcher I am today. Special thanks are also due to my dear friend Ali Mulla, who was with me from the very beginning, helping me prepare my Ph.D. applications. I would also like to thank my examining committee, Dr. Marc St-Hilaire, Prof. Aruntia Jaekel, Dr. Nancy Samaan, and Dr. Tet Yeap. Their constructive criticism and most valued feedback significantly added to my work's readability and helped shape it into its final form.

Many thanks go to my two dear friends Ahmed Abdelrehim and Ahmed Elshafei, who have helped me in many ways without even realizing it. Thanks also go to my closest friend Shady, who was keen on traveling to Ottawa to attend my thesis defense despite being overwhelmed with both work and study. Special thanks are also due to our dear family friend Dr. Maged Botros for his constant care and great support throughout my studies. Thanks for all the true friends I made along the way and for all who believed in me, especially Dr. Salma Elkawah, who convinced me to keep trying in a time I was considering giving up.

As for my family, I count them among the greatest blessings. This achievement would have not been possible without the support and encouragement of both my parents, who have been with me on the phone each and every day, making me part of their lives as they are part of mine, despite the distance and time difference. Counted among my closest family is Dr. Mahmoud Nour El-Din, to whom I will always be grateful for his words of encouragement and his belief in me. May we all meet again one day in the best of places.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ACIT	Aware Conditional Immediate Tagging
CO	Central Office
CDN	Content Delivery Network
DWDM	Dense Wavelength Division Multiplexing
DBA	Dynamic Bandwidth Allocation
DFB	Distributed Feedback Laser
DT	Dense Tagging
ED	Excess Distribution
EGP	Earliest Gap-Filling
EGF/FC	Earliest Gap-Filling or Full Credit
EGFC	Earliest Gap-Filling and Credit
EPON	Ethernet PON
FBG	Fiber Bragg Grating
FiWi	Fiber-Wireless
FS	Fixed-Slot
FS-FC	Fixed-Slot with Full Credit
FS-GF	Fixed-Slot with Gap-Filling
FS-GF/FC	Fixed-Slot with Gap-Filling or Full Credit

FS-GFC	Fixed-Slot with Gap-Filling and Credit
FTTx	Fiber-to-the-*x*
GPON	Gigabit PON
IEEE	Institution of Electrical and Electronics Engineers
IMIX	Internet Mix
IoT	Internet of Things
ITS	Intelligent Transportation Systems
ITU-T	International Telecommunications Union – Telecommunications
LC-VCSEL	Long-Cavity Vertical-Cavity Surface-Emitting Laser
LGF	Largest Gap-Filling
LGF/FC	Largest Gap-Filling or Full Credit
LGFC	Largest Gap-Filling and Credit
LR-PON	Long-Reach Passive Optical Network
LROAN	Long-Reach Optical Access Network
MCC	Mobile Cloud Computing
MEC	Mobile Edge Computing
MPCP	Multipoint Control Protocol
NG-PON2	Next-Generation Passive Optical Network (stage 2)
OLT	Optical Line Terminal
ONU	Optical Network Unit
OOB	Out-of-Band
PON	Passive Optical Network
rCDN	reverse Content Distribution Network

RN	Remote Node
RTT	Round-Trip Time
SAMAH	Signaling-Ahead for Media Access Handovers
SC-VCSEL	Short-Cavity Vertical-Cavity Surface-Emitting Laser
SIEPON	Service Interoperability in EPONs
SLA	Service Level Agreement
TDMA	Time-Division Multiple Access
TWDM	Time and Wavelength Division Multiplexing
TT	Taking Turns
TTACT	Taking Turns with Adaptive Cycle Time
UIT	Unconditional Immediate Tagging
UITR	Unconditional Immediate Tagging with Reset
UITX	Unconditional Immediate Tagging with cycle eXtension
WDM	Wavelength Division Multiplexing
XG-PON	10 Gigabit Passive Optical Network

Chapter 1

Introduction

The advent of the *internet-of-things* (IoT) together with the evolution of smart and mobile devices have brought us into a new era, in which everything is being reshaped to fit rising trends and better serve emerging applications. On one hand, network infrastructures are going through many reformations to better accommodate the exponential increase in user traffic and meet the demanding requirements of various applications. On the other hand, new networking concepts are being introduced that radically change the way services can be provided, bringing forward many new possibilities and many accompanying challenges.

With the birth of *passive optical networks* (PONs), the last-mile bandwidth bottleneck was finally alleviated. Their high capacities and low deployment costs made them one of the best broadband access network solutions that can accommodate the increasing bandwidth demand. Since then, advancements in optical technologies have been further improving their data rates and significantly extending their network reaches, allowing PONs to become an integral part of next-generation access networks and gradually shifting the last-mile bottleneck from a bandwidth problem to a computing one.

Many of the applications running on wireless resource-constrained devices generate huge amounts of traffic, which may then need to be processed on a remote device that has more powerful

resources [1]. However, with the intolerant delay requirements of some of these applications and the continuously increasing numbers of such devices, conventional offloading to a faraway cloud is no longer a viable option. Not only may it be infeasible and unaffordable to carry such huge amounts of traffic through the network [2], but the cloud itself may no longer be able to meet the strict latency requirements of many of these applications. Instead, new concepts such as fog and edge computing have risen to address these new requirements by bringing more resources to the edge of the network in order to deal with much of that traffic locally rather than having it go through the core network [3]. The integration of these new computing concepts with access networks has therefore attracted much attention.

Another major concern within access networks is their energy consumption. The growing carbon footprint of communication networks has made their energy-efficiency both an economical interest and an ecological one. This led energy conservation to be studied for access networks, especially passive optical networks, with them being one of the leading and widely deployed technologies today. Although they are considered to be the most energy-efficient among wired access technologies, their power consumption is still considerably high that they are expected to be among the largest energy consumers over the next few years [4].

In this dissertation, we tackle these topics both separately and jointly. We model the offloading performance through an analytical framework and compare it against extensive simulation results. We then analyze different factors that may affect the performance and examine various tradeoffs that are posed when energy-conserving frameworks are applied. We also propose novel bandwidth-allocation schemes that aim to maximize both performance and energy-savings. In the remainder of this chapter, we present the motivation behind our research in Section 1.1 followed by our research objectives in Section 1.2. We then present our main contributions and the organization of this dissertation in Sections 1.3 and 1.4.

1.1 Motivations

Integrating fog computing with optical access networks is believed to form a highly capable front-haul that would live up to the various requirements and challenges of tomorrow's access networks. Although a number of architectures have been proposed for integrating fog computing with optical access networks, the focus was either on the fog computational gains or the cloudlet placement problem with less or no attention to the offloading performance itself, its potential bottlenecks, and its side effects on normal traffic. Because optical access networks were not originally designed to support edge-to-edge communications nor carry offloaded traffic to a cloudlet, the network architecture and employed bandwidth allocation both need to be reconsidered in this new context. Both centralized and decentralized-based allocation paradigms need to be examined to see which has better offloading performance, especially in *long-reach passive optical networks* (LR-PONs), where the long propagation delays severely degrade the performance of centralized bandwidth allocation. Moreover, the optimal cloudlet placement needs to be investigated for each allocation paradigm as well as the side effects of offloading on regular upstream traffic. Furthermore, the pressing energy-efficiency requirement on today's communications systems adds another challenge to this integration problem. This is because energy-conserving frameworks typically result in tradeoffs between network performance and energy-saving gains. In fact, most energy-driven studies for PONs tend to sacrifice network performance, in terms of delays and throughput, in favor of energy-savings. Additionally, very few studies examined the fog integration performance under an energy-conservation framework. It is therefore essential to study both the network performance and the offloading performance under power-conservation frameworks and find novel ways to improve their performance while maximizing their energy savings.

1.2 Research Objectives

In this dissertation, we examine which bandwidth allocation scheme would be a better fit for integrating fog and edge computing with optical access networks. For that, we propose, test, and refine different allocation schemes with different cloudlet architectures. We divide the scope of our work into four phases, each with its own objectives as follows:

1. Examining the fog integration problem form a bandwidth allocation perspective:
    a. investigate how both centralized and decentralized bandwidth allocation paradigms can be modified to support offloading,
    b. find out the optimum and most economical cloudlet placement for each paradigm,
    c. develop an analytical framework that models the offloading delays,
    d. develop a discrete-event based simulation that models the offloading performance,
    e. compare the analytical framework against numerical results,
    f. examine the limitations and potential bottlenecks within each allocation scheme,
    g. study the side effects of offloading on regular traffic,
    h. determine which allocation paradigm is a better fit for fog integration.
2. Investigating the possibility of exploiting the resources of the optical access network itself and the feasibility of composing services from these resources:
    a. formulate the edge computing problem,
    b. develop a service composition framework,
    c. examine how services can be exchanged between edge units in a network that was not originally designed to support direct edge-to-edge communications,
    d. compare the service composition performance when the underlying allocation scheme is either centralized or decentralized-based,
    e. compare the advantages and limitations of each allocation paradigm.

3. Adding the energy-efficiency requirement to our bandwidth allocation schemes:
    a. add the energy dimension to the discrete-event based simulation,
    b. propose and modify allocation schemes to become sleep-aware to conserve energy,
    c. develop and test novel mechanisms to improve their network performance,
    d. examine tradeoffs between energy-saving gains and network performance.
4. Reexamining the fog integration problem form an energy-efficiency perspective:
    a. examine which allocation paradigm would have the overall best performance in terms of offloading, network performance, and energy efficiency,
    b. examine the cost of each allocation paradigm and its cloudlet placement options.

1.3 Main Contributions and Scholarly Achievements

In this dissertation, the integration of fog and edge computing with optical access networks is analyzed from the perspective of the bandwidth allocation and its energy-efficiency both analytically and through extensive simulations.

The contributions of this work can thus be summarized as follows:

- Comparing between centralized and decentralized-based offloading in LR-PONs. For this, we first study how offloading can be supported in each paradigm before developing an analytical model that identifies the key parameters that affect the offloading delay performance. The model is then compared against simulation results, which also examine the effect of various elements on the offloading performance such as the network extension, size of offloaded data, and changing the cloudlet placement.
- Proposing novel schemes that support edge-to-edge communications in LR-PONs and investigating the possibility of composing certain services using the devices and the available resources of the optical access network itself.

- Examining the integration problem from an energy-efficiency perspective, for which we develop novel energy-efficient allocation schemes that aim to maximize network performance. We also study the tradeoffs between the energy-saving gains and performance to determine which allocation paradigm would have the overall best performance.

From the aforementioned contributions, the following publications have been produced:

Journal/Magazine Publications:

- A. Helmy and A. Nayak, "Integrating fog with long-reach PONs from a dynamic bandwidth allocation perspective", *IEEE/OSA Journal of Lightwave Technology*, vol. 36, no. 22, pp. 5276-5284, Nov. 2018. (Chapter 3)
- A. Helmy and A. Nayak, "Towards parallel edge computing in long-reach PONs", *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 9, pp. 736-748, Sep. 2018. (Chapter 4)
- A. Helmy and A. Nayak, "Energy-efficient decentralized framework for the integration of fog with optical access networks", *IEEE Transactions on Green Communications and Networking*, under review. (Chapters 5 and 6)
- A. Helmy and A. Nayak, "Centralized vs. decentralized bandwidth allocation for supporting green fog in next-generation optical access networks", *IEEE Communications Magazine*, under review. (Chapters 3, 5, and 6)

Conference Proceedings:

- A. Helmy & A. Nayak, "Decentralized bandwidth allocation framework for energy-efficiency and fog integration in PONs", *IEEE International Conf. on Computing, Networking, and Communications (ICNC)*, Hawaii, USA, Feb. 2020. (Chapter 5)

- A. Helmy and A. Nayak, "Towards more dynamic energy-efficient bandwidth allocation in EPONs", *IEEE Global Communications Conf. (Globecom)*, Abu Dhabi, UAE, Dec. 2018. (Chapter 5)
- A. Helmy and A. Nayak, "Towards green fog-LR-PON integration for wireless backhauls", *IEEE Globecom*, Abu Dhabi, UAE, Dec. 2018. (Chapter 6)
- A. Helmy, N. Krishna, and A. Nayak, "On the feasibility of service composition in a long-reach PON backhaul", *IEEE 22nd Conf. on Optical Network Design and Modeling (ONDM)*, Dublin, Ireland, May 2018. (Chapter 4)
- A. Helmy and A. Nayak, "Fog integration with optical access networks from an energy efficiency perspective", *IEEE International Conf. on Computer Communications (INFOCOM)*, submitted. (Chapters 5 and 6)

1.4 Dissertation Organization

The remainder of this dissertation is organizing as follows. Chapter 2 presents a general background on PONs and fog computing along with related works from the literature. In Chapter 3, we focus on the integration of fog computing with optical access networks from a bandwidth allocation perspective. In Chapter 4, we explore edge computing and service composition in long-reach optical access networks by using the resources of the network itself. In Chapter 5, we shift our attention to the energy-efficiency challenge in PONs, where we propose new allocation schemes that significantly add to the network performance with very negligible energy sacrifices. In Chapter 6, we compare between the two allocation paradigms in terms of both energy-savings and network performance to determine which has the best overall performance. Chapter 7 concludes our study and presents possible directions for future research. Finally, we shed some light on the work done to model and simulate passive optical networks in Appendix A.

Chapter 2

Background and Related Work

With the emergence of the *internet-of-things* (IoT) and the continuous widespread of smart devices, the Internet is constantly being drawn out from fixed stations within offices and buildings to our very surroundings. This huge shift has been accompanied by the rise of many new demanding applications and a correspondingly huge increase in internet traffic, which is expected to continue on growing to reach hundreds of exabytes per month in the next few years [5]. Moreover, global mobile data traffic is now growing nearly twice as fast as fixed traffic such that wireless and mobile devices are expected to account for 71% of total IP traffic by 2022 [6].

Conventional ways of dealing with this rapid traffic growth have always been through improving communication infrastructures and providing higher data rates. To that end, optical networks have taken a major role in serving as both back and front-haul networks [7]. *Passive optical networks* (PONs) in particular have played a central role [8]. Their high capacities and data rates combined with their long reaches and wide coverages have made them a very popular choice for serving wireless infrastructures and an integral part of next-generation access networks. Advancements in optical technologies have enabled PON data rates to be increased above 10Gbps with the next-generation 100Gbps-PONs currently being standardized [9]–[11]. However, the nature of access network traffic has significantly changed. New requirements and services now require much closer computing resources than those offered by the cloud in order to provide less

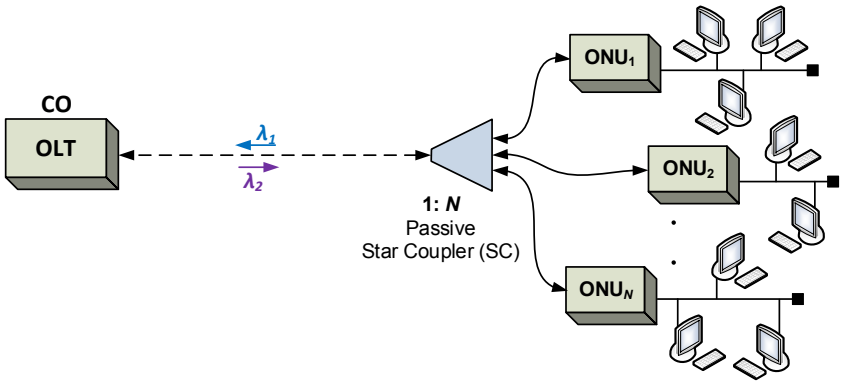


Figure 2.1: PON's tree architecture.

latency, more security, and better location awareness. The last-mile bottleneck has therefore gradually shifted from a bandwidth problem to a computing one.

In Section 2.1, we give an overview of both PONs and fog computing, before discussing related works on their integration and on energy conservation in PONs in Section 2.2.

2.1 Background

2.1.1 Passive Optical Networks (PONs)

When PONs were first introduced, their high capacities and low deployment costs soon made them one of the best access network solutions. Their low deployment costs were mainly due to having no active elements in the field. The only active elements resided either at the *central office* (CO), i.e. the *optical line terminal* (OLT), or in the users' premises, i.e. the *optical network units* (ONUs). Although a PON can be based on either one of many topologies, the most commonly used is the tree topology, in which the main fiber is split into many branches using a 1: *N* passive star coupler to serve many locations in the access zone, as illustrated in Figure 2.1. Fiber splitting may also be carried out in multiple stages, which may be optimal for customers that are clustered in several separate areas. However, each additional split stage increases the power loss by

approximately 1–1.5 dB due to the loss of the additional connectors and the uneven division of power in each splitter [12].

There have been many PON standards, which either belong to *the Institution of Electrical and Electronics Engineers* (IEEE) or *the International Telecommunications Union – Telecommunications Standardization Sector* (ITU-T). The former had PON technologies standardized under the Ethernet portfolio; IEEE 802.3ah for *Ethernet PON* (EPON) and IEEE 802.3av for 10G EPON), with a focus on high speed and low cost. The latter, on the other hand, was dominated by government representatives and tended to be more expensive with delayed availability [12]. ITU has standardized *Gigabit PON* (GPON) and *10 Gigabit PON* (XG-PON). ITU has also standardized a *next-generation PON stage 2* (NG-PON2), which employs *time and wavelength division multiplexing* (TWDM), with a rate of 40Gbps, but still compatible with preexisting GPON and XG-PON standards.

By providing fiber all the way to the access zone, the last-mile bandwidth bottleneck problem was greatly alleviated. PONs' particular topology, however, which is different in the upstream direction from that in the downstream direction, posed some challenges in various domains such as upstream bandwidth allocation, the security of downstream communications, energy-efficiency, and fault tolerance. These different challenges gained much research interests over the past two decades and produced many significant contributions to the literature.

2.1.1.1 **Long-Reach PON (LR-PON)**

Long-reach PONs are widely considered as a highly integrated and cost-effective version of PONs. By exploiting several optical technologies, they combine access and metro networks into a single integrated network saving enormous operational and capital costs. They may also extend the coverage of the network by tens of kilometers up to a hundred or more [13]–[16]. The concept

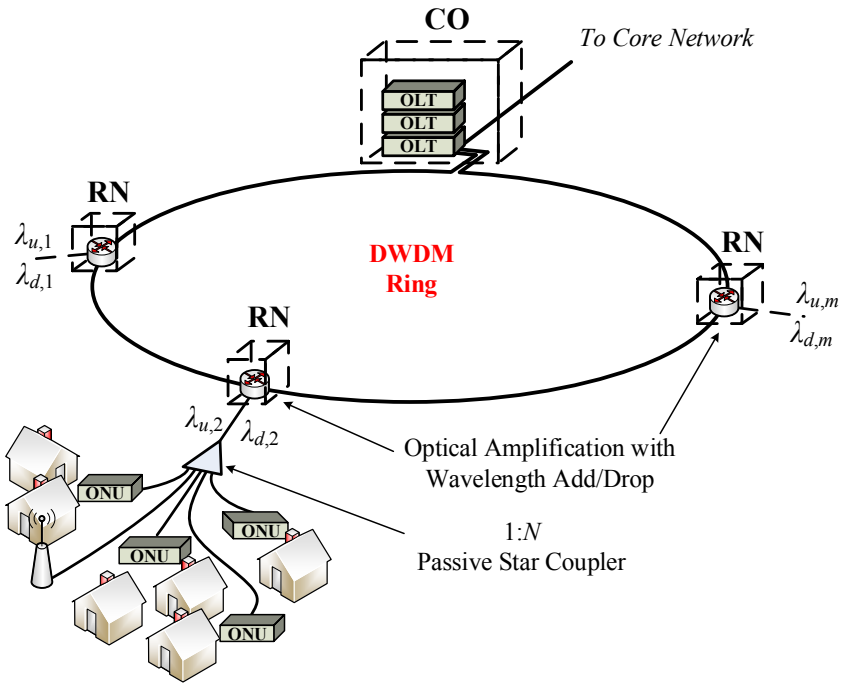


Figure 2.2: LR-PON ring-and-spur architecture.

of a LR-PON has been proven experimentally for a 10Gbps symmetrical network with a 100km reach and a 1024-split [17]. The term LR-PON, however, has frequently been debated since, in most cases, only the access portion of the network remains passive, whereas some amplification may be required at the *remote nodes* (RNs). For such reason, the technology has been widely, and more accurately, dubbed as a *long-reach optical access network* (LROAN).

Figure 2.2 illustrates one of the most common LR-PON architectures, adopted from [18], where a single central office can be connected to many access zones via a metro ring through *dense wavelength division multiplexing* (DWDM). In such architecture, each access zone is served with dedicated upstream and downstream wavelengths that enable communications with its associated OLT. The ONUs in each access zone can then either provide wired services for residential or business subscribers, through what is known as *fiber-to-the-x* (FTTx) technologies, or may be part of a wireless infrastructure.

2.1.1.2 **Dynamic Bandwidth Allocation (DBA)**

The logical connection between the OLT and ONUs is different on the downstream from it on the upstream. In the downstream, the network forms a point-to-multipoint topology. Therefore, all downstream transmissions are typically done in broadcast-basis. Contrarily, the network forms a multipoint-to-point topology in the upstream, where multiple ONUs transmit toward their associated OLT through a shared optical wavelength. Because ONUs are unable to listen to each other's transmissions, due to the directional property of the coupler, some media access control is needed to avoid data collisions and enable efficient utilization of the shared upstream medium.

The simplest way to regulate ONUs' transmissions on a common assigned wavelength would be to statically assign timeslots to the ONUs fixing their allocated bandwidths. However, it has been shown in the literature how access network traffic exhibits a high degree of 'burstiness' that is long-range dependent following a heavy-tail distribution [19]. This bursty nature of access traffic has a severe impact on the performance of static timeslot assignments [20]. It may lead to some ONUs using very little of their allocated bandwidth while other heavily-loaded ONUs are in desperate need of additional bandwidth. Another issue with static timeslot assignments arises when the traffic is composed of variable-sized packets, as the case in EPONs. With a fixed timeslot assignment, ONUs would then usually leave unused remainders at the end of their timeslots that do not fit in any of their queued packets. For these reasons, *time-division multiple access* (TDMA) has been adopted in most PON standards for enabling *dynamic bandwidth allocation* (DBA).

The EPON standard bases itself on the *multipoint control protocol* (MPCP), which mainly uses two control frames; GATE and REPORT for setting up the network and enabling OLT-ONU communications. A GATE message is used by the OLT to convey certain network parameters to the ONUs such as those used to synchronize them to the OLT's clock or initiate their auto-

discovery. The GATE message is also used by the OLT to grant each ONU a transmission window (also referred to as a variable timeslot) by specifying its length and start time. On the other hand, REPORT messages are used by ONUs to report certain information to the OLT, such as their current upstream buffer status, which indicates how much bandwidth they require for transmitting the traffic accumulated from their connected users. Because GATE and REPORT messages both have reserved bits and unused padding, they can be tailored to carry more information as needed. That way, MPCP forms a foundation layer that is necessary for network operations but is still extendable and reconfigurable allowing many bandwidth allocation algorithms to be realized on top of it. Bandwidth allocation algorithms can then be classified under centralized or decentralized-based according to where the media access management takes place.

2.1.1.2.1 Centralized bandwidth allocation

In centralized bandwidth allocation, the OLT arbitrates bandwidth allocation among the ONUs based on their received reports. Its realization in EPON is straightforward with MPCP. As illustrated in Figure 2.3, the OLT typically polls each ONU with a GATE message, granting it a window to transmit, according to the information received in its previous REPORT message. The OLT is thus able to adapt transmission window sizes of ONUs in accordance with their bandwidth demands, thereby making the bandwidth allocation dynamic. The ONUs, on the other hand, do not need to monitor the network state nor negotiate new parameters, which makes their design relatively simple. The drawback of centralized allocation, however, arises from the non-zero ONU-OLT propagation delays, which prevents the OLT from having real-time information of the ONUs' traffic demands, making it less responsive. A PON with centralized DBA is therefore considered to be a remote scheduling system that suffers from a control-plane delay proportionate to the distances between the OLT and the ONUs. This delay becomes more severe in LR-PONs,

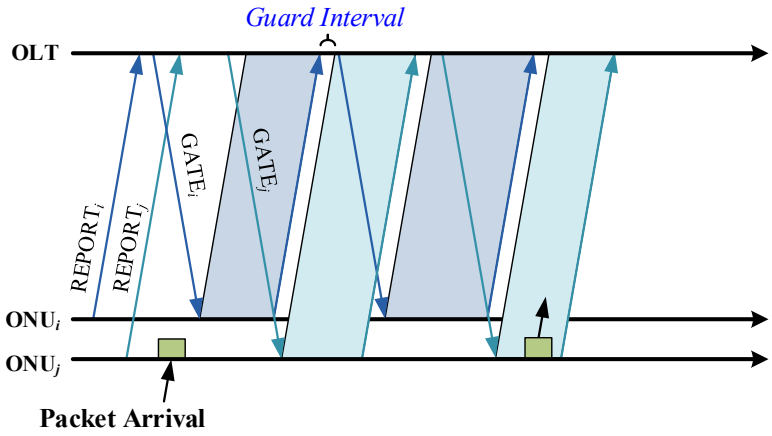


Figure 2.3: Centralized DBA using polling.

where the *round-trip time* (RTT) may reach up to 1ms, causing the centralized bandwidth allocation performance to be ultimately degraded. To address this issue, many centralized allocation schemes were proposed and studied in the literature [21]. Some proposed estimation-based techniques to predict ONUs' demands, such as [22] and [23], whereas others used credit-based schemes, in which ONUs are granted slightly more bandwidth than requested to account for newly arriving traffic [24]. Following studies suggested using multiple threads of communications between the OLT and ONUs [18], which inspired several related works [25]–[28].

2.1.1.2.2 Decentralized bandwidth allocation

Decentralized bandwidth allocation was introduced as another way of dealing with the degradation in centralized bandwidth allocation due to the long propagation delays of LR-PONs. In such schemes, the ONUs themselves manage their upstream media access instead of having to report their buffer status to a remote OLT and then wait for a grant to transmit. This allows packet pre-transmission delays to become independent of the ONUs' distances from the OLT and thus achieves better performance in LR-PONs. The challenge however in such schemes is acquiring means of inter-ONU communications.

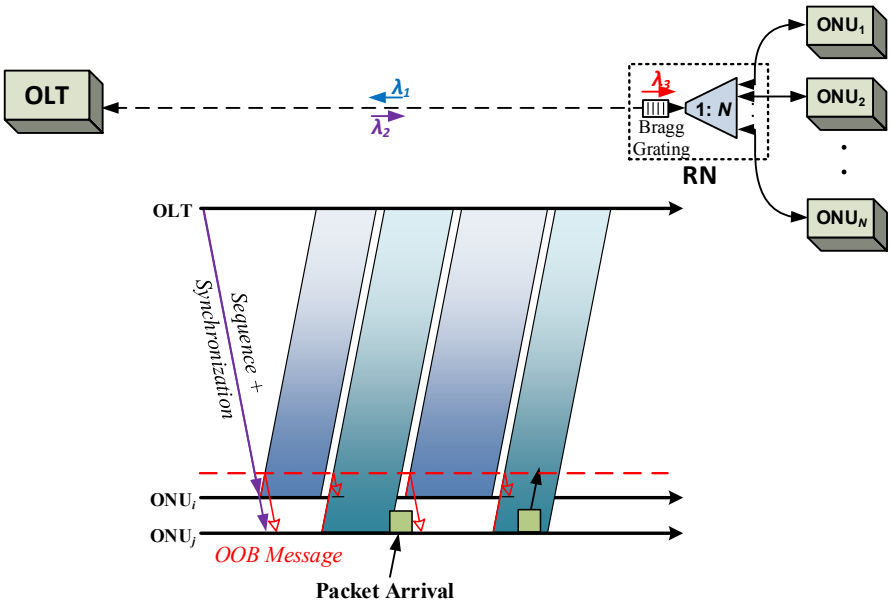


Figure 2.4: Decentralized TTACT using OOB tagging.

In [29], a decentralized scheme was proposed to handle upstream traffic in LR-PONs, in which communications between ONUs were enabled by placing a *fiber Bragg grating* (FBG) near the RN. This allowed a specific wavelength to be selectively reflected back, creating an *out-of-band* (OOB) multipoint-to-multipoint network among the ONUs. This inter-ONU communications concept was extensively studied in [30] and was found to be a viable option for facilitating inter-ONU networking. Using this OOB channel, the ONUs in [29] were able to take turns transmitting on the upstream by using this channel to announce to each other the lengths of their transmissions without exceeding a certain maximum. Since the upstream cycle adapted to the ONUs' transmissions, the scheme was called *Taking Turns with Adaptive Cycle Time* (TTACT).

Figure 2.4 illustrates the concept of TTACT, where it can be seen how chances of upstream inter-transmission gaps are reduced since the time it takes an ONU to inform the following ONU of its transmission duration (on the control channel) would be during its ongoing data transmission (on the upstream channel). Therefore, in this decentralized scheme, each ONU virtually grants

media access to the following ONU by reporting to it how long its transmission will last. With no negotiations with the faraway OLT, the packet delays in TTACT are different in nature from those in centralized polling since they depend on the propagation distances between the ONUs themselves (through the reflective device) rather than their distances from the OLT.

Although TTACT requires some architectural modification (i.e., adding a physical-layer reflective device), it can still make use of the Ethernet MPCP framework. For instance, the OOB messages exchanged between the ONUs, which are called tag frames [29], are similar in structure to that of an MPCP REPORT. Moreover, even though the OLT, in this scheme, is demoted to a supervising role, it still uses GATE messages to initially, and occasionally, set up some parameters such as the optimal ONU transmission sequence and the maximum window sizes. The scheme thus maintains some form of centralized control that enforces users' *service level agreements* (SLAs).

Another decentralized approach was proposed in [31], where inter-ONU communications were enabled through a reflective grating array by using two upstream wavelengths. The scheme required placing multiple tunable transmitters in the ONUs along with a tuned array of receivers to be able to use all channels simultaneously. This, however, makes ONUs more complex and more expensive. It also requires coordinating transmissions on two upstream channels instead of one. Additionally, an even number of ONUs (4 ONUs) was assumed in order to give each pair a dedicated upstream wavelength for both inter-ONU communications and upstream transmissions. This would be difficult to realize with a larger number of ONUs or with an odd number of them. Finally, the setup of this network and the management of user SLAs were not addressed in [31].

In our work, we choose TTACT as the candidate decentralized scheme and tailor it to facilitate fog and edge computing by developing new tagging schemes in Chapters 3 and 4. We also use the same inter-ONU communications concept to develop a decentralized energy-efficient framework in Chapter 5.

2.1.1.3 **Energy Conservation in PONs**

Even though PONs are considered to be the most energy-efficient among wired access technologies, energy-efficiency is still one of their main design requirements and a growing area of research [32]. Part of this may be due to the realization that access networks consume around 70% of overall Internet energy consumption, which significantly contributes to the operational expenditures of telecommunication networks [33]. This, coupled with the wide deployment of FTT*x* technologies, which has surpassed an estimated household penetration of 50%–75% in modern countries [34], [35], has raised many environmental concerns due to the resulting carbon footprint [36]. In fact, it was estimated that the global annual PON power consumption has exceeded 2.37TWh, which is sufficient to power 300,000 households with more than 1 million people. Within PONs themselves, the ONUs have been found to be responsible for 60-88% of the power consumption [36], [37]. This can be attributed to the downstream broadcasting nature of PONs, which forces the numerous ONUs to continuously listen to the OLT's downstream transmissions and examine each of its frames. Many strategies have therefore been proposed in the literature to reduce their power consumption by focusing on the ONUs.

The first works calling for more energy-efficient PONs were published nearly a decade ago, in which power saving was proposed to be accomplished via introducing a sleep mode for ONUs, partially turning them off to conserve power [38]. The concept had already been adopted in wireless communications and was beginning to be studied for wired optical networks [39]. This however presented some challenges in PONs since downstream transmissions are done in broadcast-basis, requiring each ONU to stay awake and continuously listen to downstream transmissions in order to filter out any relevant frames. Moreover, for an ONU to switch back from sleep-mode, a warm-up period is required, during which the ONU must resynchronize to the OLT's clock. This warm-

up overhead can be as long as 2–5ms, making it necessary to elongate the cycle duration in order to achieve significant energy savings. This, however, results in longer packet delays and a corresponding degradation in network performance. The ONU cyclic sleep mode has then been outlined in current standards, namely the *Service Interoperability in EPONs* (SIEPON) and the ITU-T series-G supplement 45 for power conservation in GPONs. Although the standards define different power states and their necessary control messages, the sleep duration and the parameters/conditions that trigger it are left outside of their scope, leaving room for various frameworks to be realized and allowing much operator flexibility.

2.1.2 Fog and Edge Computing

In the past decade, most mobile applications have been following the traditional client-server model, by which an application is typically split into a front-end client program running on the device and a back-end server program running at the cloud datacenter. This model is the basis of *mobile cloud computing* (MCC), which enables the mobile device to utilize remote cloud-based computing and storage resources. With the advances made in wireless infrastructures, enabling them to be more reliable and support higher data rates, the client-server model soon became more feasible and very common for many mobile applications [40].

Though the use of centralized cloud datacenters offers virtually unlimited resources, it also leads to a large separation between the mobile device and these resources. End-to-end communications between both parts of the mobile application then usually involve many network hops resulting in high latencies and adding to network congestions. With the increasing numbers of devices and with the introduction of more demanding applications that cannot afford such latencies, fog and edge computing soon became a necessity to provide better *quality of service* (QoS) and support many emerging applications. These computing concepts have also become

essential for enabling the IoT era and the so-called 'Tactile Internet', which aims to have millisecond latencies.

Both fog and edge computing concepts revolve around moving the processing closer to where the data is being created and harvesting the vast amount of the idle computation power and storage space distributed at the network edges. While both terms are usually used interchangeably, there are subtle differences between the two. Fog computing is a term originally created by Cisco, which considers it to be a decentralized computing concept that still relies on the central cloud to carry out complex processing [41]. It is a relatively broad concept, where its computing can be located anywhere between the large-scale cloud data-centers and the end devices [42]. In fact, any device that has computation, networking, and storage capabilities can potentially act as a fog node and offer its idle computing resources for serving other devices [43]. This includes set-top boxes, switches, routers, base stations, proxy servers or any other network devices [41]. The challenge is then to manage these devices and translate the various services within such a heterogeneous computing environment. Because fog computing complements the cloud without replacing it [44], it is considered to be a cloud add-on that extends some of its capabilities to the edge, enabling tasks to be processed in closer proximity [45].

Edge computing, on the other hand, can exist as a standalone approach and, by definition, limits the location of the computing resources to the network edge, usually within the edge nodes themselves. It is also motivated by moving 'the code' towards 'the data', especially when transferring 'the data' to 'the code' is either too slow or too expensive [46]. The term is often associated with mobile networks, where *mobile edge computing* (MEC) servers are deployed at base stations to provide processing and storage capabilities at the edges of cellular networks [41]. This allows computationally-intensive and latency-critical tasks to be offloaded from resource-constrained mobile devices and carried out within close proximity in a timely fashion. The edge

servers, however, may still be connected to cloud datacenters in remote locations, to forward some tasks that cannot be processed at the edge server.

The term fog computing is therefore much broader [1], making 'the edge' part of 'the fog'. Still, because both fog and edge resources are much closer to the devices than the cloud, they are both able to overcome the cloud's high latency problem and satisfy the stringent delay requirements of many demanding applications [47]. Both computing concepts are also driven by the increasing number of connected devices and sensors since more resources can be deployed to cope with the increasing traffic load and greatly alleviate the pressure on the core network. Because deploying new data-centers can be very cost-prohibitive, fog and edge computing are considered to be more scalable than the cloud. Moreover, fog and edge computing provide better location awareness, higher security, and more reliability.

The term *cloudlet* is another term that appears in some related studies to be something different when it is essentially the same. A cloudlet is typically a lightweight cloud server or micro-datacenter that is placed near the edge of the network to carry out offloaded tasks. Because it may sit anywhere between the cloud and the device, it falls under the fog's territory. A cloudlet can therefore be considered as a special case of fog computing, where a dedicated server is added to the network. Table 2.1 highlights the main differences between these similar computing concepts showing how some of them may be subsets of others.

The ubiquity of mobile devices and their evolution into significant service computing platforms has also given way to new possibilities, where devices in the same vicinity can start exchanging services among themselves. This new vision has led to developing various service discovery and service composition techniques in order to allow individual services, in a service-oriented environment, to be looked up and properly combined to solve more complex tasks [48]. While this environment can be anything from a mobile network to a deep space research station,

Table 2.1 Different Computing Paradigms and Applications

	Fog			
		Cloudlet	Edge	
				MEC
Cloud connection	yes	yes	yes or no	yes or no
Main computational element	any device with sufficient resources	micro-datacenter or server	micro-datacenter or server	MEC server
Types of users	mobile/ stationary	mobile/ stationary	mobile/ stationary	mobile
Types of applications	various	various	mobile/ IoT	mobile

the service itself can be any accessible software component, hardware resource, or any data segment that can be offered to another device [49]. This new concept has brought many potential assets for a wide range of applications and scenarios such as image processing, sharing GPS/internet data, crowd computing, social networking, or aggregating and integrating sensor data to discover meaningful trends such as current weather or traffic conditions [50]. While some studies focused on the device-mobility aspect of service composition and its feasibility in wireless networks [51]–[53], other studies considered it from an energy consumption perspective [54]. Yet, the feasibility of service composition in optical access networks has not yet been examined.

2.1.2.1 **Fog and Edge Computing Applications**

The areas of fog and edge computing are overlapping and both terminologies are frequently used interchangeably. Although these computing concepts appear to be new, some believe their roots reach back to the late 1990s, when *content delivery networks* (CDNs) were first

introduced [55]. A CDN employs edge nodes close to users to pre-fetch and cache popular or anticipated web content from content sources and then make it readily available. This considerably reduces the delays associated with accessing the content and results in substantial bandwidth savings, especially when the content is video. The edge nodes can also perform some content customization, such as adding location-relevant advertising. Fog and edge computing can therefore be regarded as an evolution of CDNs with a wide spectrum of applications.

Today, fog and edge computing have mainly been tied to IoT devices and the next generation of mobile networks. As IoT devices are very resource-constrained, they have to rely on other resources to enhance their capabilities. Fog and edge computing are therefore essential for carrying out their computationally intensive tasks as well as prolonging the lives of their batteries, which are usually difficult to recharge or replace [1], [40]. One the other hand, the next generation of mobile networks, i.e. 5G, is to offer much lower latencies in delivering services compared to existing cellular technologies [43], [56]–[58]. This can only be achieved by having local resources assist in providing real-time and interactive services rather than depend on remote cloud resources.

The proximity of these computing resources therefore helps in four main ways [55]. The first is being highly responsive to delay-sensitive applications by providing low end-to-end latency. The second is reducing network congestions and providing more scalability via edge analytics. In other words, instead of using high bandwidth to send raw data, it can be analyzed at the edge before sending much smaller extracted information to the cloud. The third is the improved security since processing the data locally is more secure than sending it over the core network to remote servers. Moreover, in case some data is to be sent, an edge server will be the first point of contact in the infrastructure and can enforce the privacy policies of its owner prior to the release of the data to the remote cloud. Finally, fog and edge computing can mask cloud outages due to

network failure, cloud failure, or denial-of-service attacks. In the following, we view some of the applications that greatly benefit from such resources.

2.1.2.1.1 Applications generating huge amounts of data

The full potential of fog and edge computing is greatly demonstrated with applications that generate huge amounts of data that would otherwise result in a substantial burden on the core network. Examples include smart grids and *intelligent transportation systems* (ITS) [59], which are both vital components of the smart city. For instance, smart grids utilize the IoT to improve and enhance the energy consumption of houses and buildings. They employ a network of millions of smart meters, sensors, and actuators to help power suppliers control and manage different resources, improve offered services, as well as cope with grid fluctuations and varying energy demand in real-time. While some data collected by meters and sensors must be sent to the control center in real-time, other data types are huge in size and more delay tolerant. For example, the U.S. smart grid is expected to generate 1000 petabytes of data each year [1]. Sending all this data to the cloud will require huge networking resources and may also often be unnecessary or prohibited due to regulations and data privacy concerns. In such cases, it is preferred to store and process the data locally rather than at the cloud.

2.1.2.1.2 Applications requiring data analysis

Other applications may additionally involve some form of analysis as part of their operation, such as facial or license plate recognition and other video surveillance applications [40]. Augmented reality is another type of application, which also requires analyzing real-world surroundings and supplementing it with additional information in real-time. Not only do some of these applications generate video streams that would consume a great amount of network

bandwidth in order to be processed at a central cloud, but their object detection and classification algorithms themselves may also be too computationally complex to be performed by the low-cost sensory devices. In such cases, fog and edge resources can perform data aggregation as well as preliminary on-the-fly analysis, before sending the partially processed data to the cloud for further processing, as opposed to sending the raw data itself to the cloud [59].

2.1.2.1.3 Applications having both features

Some applications generate huge amounts of data and, at the same time, disparately require timely analytics. An example of such applications is connected vehicles. The motivation behind connecting vehicles is to increase safety, reduce traffic congestion, and provide new opportunities for numerous value-added services such as the car finder and parking location services [40]. Yet, these vehicular networks generate huge amounts of raw data. For instance, a connected vehicle is estimated to produce tens of megabytes of data per second, which conveys information about its route, speed, operating conditions, surrounding environment, and videos recorded by the vehicle's safety cameras [1]. An autonomous vehicle, on the other hand, will generate even more data, which is estimated to reach one gigabyte per second [1]. Such huge amounts of data need to be analyzed so that decisions can be made instantly. However, the latency requirements of this technology cannot be met with existing vehicle clouds, which have end-to-end latencies between 100ms to 1s. Placing edge resources at roadside base stations are therefore essential for receiving and analyzing messages from proximate vehicles and roadside sensors before responding with hazard warnings and latency-sensitive messages within a 20ms end-to-end delay. In several research works the resources of the vehicles themselves were proposed to also be used as part of the fog to assist with the edge computing [60], [61].

2.2 Related Work

2.2.1 Fog Computing in PONs

Both optical access networks and fog computing share the same goal of addressing user traffic and its continuous growth from different perspectives. While the former tends to address the growing bandwidth demand by offering high transmission rates, the latter looks into the traffic's underlying requirements stemming from its corresponding applications. It is therefore not surprising that there have been several architectures in the literature envisioning their integration together.

One of the first studies on integrating computing resources with PONs can be found in [62], in which cloud resources are proposed to be brought over to the wireless frontend of access networks rather than reside behind the optical backhaul. In [63], long-reach PONs are connected with fog micro-datacenters that are optimally placed near RNs to serve end devices. In [64], cloudlets are brought even closer to end devices by directly connecting them to the ONUs in a *fiber-wireless* (FiWi) setting. In [65], it is proposed that the PON itself may provide the storage and processing capabilities needed by making use of the resources of the OLT together with the numerous ONUs. The PON is then seen as a virtual network of servers interconnected by high-speed fiber links.

In other studies that considered fog's integration with optical access networks, the focus was usually on the fog computational gains or the cloudlet placement. The studies, however, paid less or no attention to examining the performance of offloading to the cloudlet, from the underlying bandwidth allocation perspective, nor to the potential bottlenecks and side effects on normal traffic. Moreover, no study examined the possibility of using decentralized bandwidth allocation to support fog and edge computing, even though these computing concepts themselves are considered to be decentralized computing paradigms.

In [66], the authors compared offloading to a MEC server (connected to the ONUs) against offloading to the centralized cloud, in a FiWi network setting. Conventional centralized-based bandwidth allocation was proposed to carry both types of traffic, but the simulated network span was too short (~20km), which does not reflect the potential degradation in the centralized bandwidth allocation performance that takes place when the span reaches hundreds of kilometers. The authors also assumed an architecture, in which each ONU is connected to its own dedicated MEC server, which may not be feasible nor economical in a large-scale network, where the ONUs are large in number.

In [67], the authors consider the cloudlet placement as an optimization problem that aims to minimize installation costs under QoS latency constraints. For that, cloudlets were allowed to be installed in the field, at the RN, or at the central office. However, bandwidth allocation was not part of the study, whereas cloudlet placements were solely based on randomly generated population densities. Moreover, field cloudlets, in their hybrid architecture, required deploying dedicated fibers and line cards for each connected ONU, which entails considerable costs.

In Chapter 3, we focus on the integration of fog computing in optical access networks, whereas, in Chapter 4, we examine the feasibility of edge computing through the resources of the network itself, looking at the advantages and disadvantages of both centralized and decentralized-based service composition.

2.2.2 Energy Conservation in PONs

The research on energy saving in PONs can be divided into two main approaches; a *physical layer approach* and a *data-link layer approach*. The former approach may use hardware modifications and modulation techniques to conserve energy. Because ONUs are the major target for energy saving in PONs, most efforts in this field have focused on the ONU's circuitry.

Examples include modifying receivers to reduce the wake-up overhead [39], introducing low-power transmitters [68], [69], and enhancing amplification [70]. Electronic component integration is another direction that can shorten the recovery time while reducing the ONUs' consumption [71]. Exploiting new modulation techniques was also proposed for faster clock recovery with no architectural modification to the ONU's receiver [72]. Other efforts proposed the employment of *adaptive link rate control* (ALRC) to conserve energy by bringing down the link rate from 10Gbps to 1Gbps under light downstream traffic [73].

On the other hand, data-link layer techniques resort to manage switching off components and functions as well as developing bandwidth allocation schemes that are aware of the ONUs' cyclic sleep mode. The overall performance then becomes dependent on the sleep duration together with the employed sleep-aware bandwidth allocation, usually resulting in a trade-off between power savings and network performance. While frequent wake-ups help reduce packet delays, such short cycles result in poor energy efficiency, especially with the imposed wake-up overhead, which was reported in [74] to be around 2ms at best. The cycle duration is therefore one of the main parameters in PON's energy management. Some studies used adaptive cycles [75], [76], whereas others leaned towards long fixed cycles to allow for sufficient sleep durations [77], [78]. Adaptive cycles were also combined with a separate control message exchange between each ONU and the OLT, at the end of each slot, which elongates the cycle enough to allow for viable sleep durations [79]. Cycle durations up to 175ms and 250ms have been used in [80] and [81], respectively, whereas conventional bandwidth allocation schemes used cycle durations up to 5ms (see for example [18], [29], [82]). This great cycle extension had negative effects on the packet delays, yet, at the same time, 5ms cycles were no longer viable since the wake-up overhead would almost be half the cycle. The sleep cycle may also be decoupled from the polling cycle, as was done in [74], where the ONU was allowed to stay in sleep mode for many cycles (up to 250 cycles

of a 1ms cycle). However, besides taking at least 4 cycle durations to fully wake up and start transmitting again, this long deep sleep may result in severe frame losses and extreme delays under heavily-loaded network conditions. The sleep duration should therefore be determined based on current network conditions as well as the application's delay constraints.

Regardless of the chosen cycle duration, almost all studies converged on fixed-slot allocation, in which both upstream and downstream transmissions for each ONU are confined within a dedicated slot to allow it to sleep for the remainder of the cycle [75]–[79], [81]. Besides enabling the ONUs to switch to sleep mode outside their designated slots, fixed-slot allocation also ensures fairness by dividing the cycle among ONUs according to their SLAs. However, to ensure sufficient time for ONUs to switch back and forth from sleep mode, even under light loads, the cycle must be fixed at its maximum duration, which reduces the bandwidth allocation from being dynamic to become static. Fixed-slot allocation then suffers from two main drawbacks; poor utilization and no allocation differentiation between heavily-loaded and lightly-loaded ONUs causing heavily-loaded ONUs to experience longer delays and potential frame losses.

In Chapters 5 and 6, we focus on the energy-efficiency challenge in PONs. In Chapter 5, we propose new allocation schemes that aim to conserve energy while preserving the network performance to a high degree. In Chapter 6, we reexamine the fog integration problem from an energy perspective comparing between centralized and decentralized performances in terms of both offloading and energy efficiency.

Chapter 3

Fog-LR-PON Integration:

a Bandwidth Allocation Perspective

A mentioned in previous chapters, integrating fog computing with optical access networks is believed to form a highly capable front-haul that will live up to the various requirements and challenges of tomorrow's access networks. However, because optical access networks were not originally designed to support fog integration nor carry offloaded traffic, the network architecture and employed bandwidth allocation both need to be reconsidered. In this chapter, we examine the key design considerations and challenges that may face such integration from a bandwidth allocation perspective. To that end, we explore decentralized-based offloading in contrast to that enabled by conventional centralized allocation, in a *long-reach passive optical network* (LR-PON), to see which has better performance. We also examine the optimal cloudlet placement for each paradigm as well as the effect of offloading on regular upstream traffic. In our study, we develop an analytical framework that models the offloading delays in each paradigm and compare it against simulation results. The contributions of the work done in this chapter are compared against the contributions of other related works in Table 3.1. Additionally, Table 3.2 summaries the notations used in this chapter along with their description and default simulation values.

Table 3.1 Comparison of PON-Fog Offloading Contributions

	[64], [66]	[83]	[67]	[29]	**This Study**
DBA	centralized	centralized	centralized	centralized vs. decentralized	centralized vs. decentralized
Network reach	20km	20km	<5km	100km	100km
Network type	FiWi	FiWi	PON	LR-PON	LR-PON
Fog-Support	yes	yes	yes	no	yes
Cloudlet placement	each ONU	ONU/OLT	hybrid	no	ONU/RN/OLT
Analytical framework	yes	no	no	no	yes
Effect on regular traffic	no	no	no	no	yes

Table 3.2 Symbol Definitions and Simulation Values

Symbol	Description	Value
$L_{\max}$	Maximum network span	100km
$L_{i,SC}$	ONU$_i$'s distance from splitter/combiner	0–5km
L_{SC}	Splitter distance from OLT (feeder span)	95km
R_{down}	Downstream transmission rate	10Gbps
R_{up}	Upstream transmission rate	1, 10Gbps
R_{OOB}	Out-of-band channel transmission rate	1, 10Gbps
R_{user}	Connected users' upstream rate	100Mbps, 1Gbps
N	Number of ONUs	16
Q	ONU buffer size	10MB
L	Normalized network load	0.1–1.1
$C_{\max}$	Max. cycle duration	5ms
T_g	Guard interval	5µs
L_{off}	Size of offloaded data	1–11MB

3.1 Network Architecture

Figure 3.1 illustrates the network architecture of a LR-PON, where the *central office* (CO) is connected to many access zones via a *wavelength division multiplexing* (WDM) metropolitan ring. As mentioned earlier in Chapter 2, each access zone is served with dedicated upstream and downstream wavelengths that enable communications with its associated *optical line terminal*

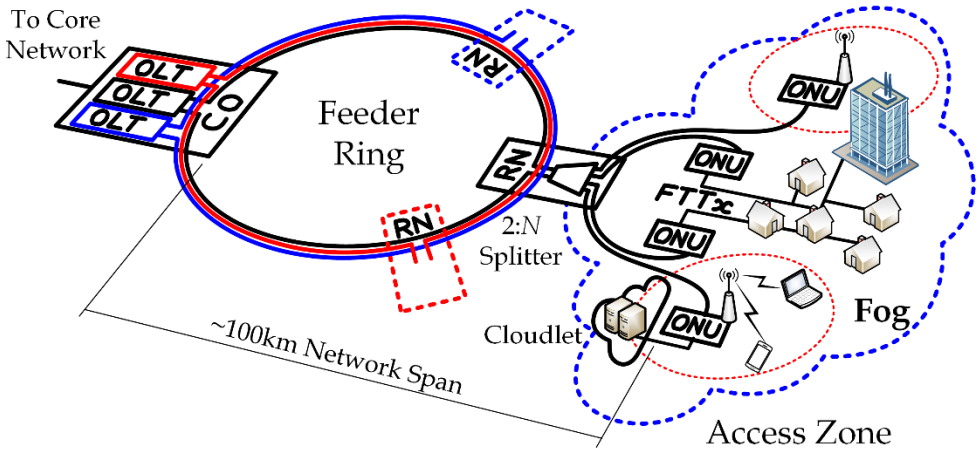


Figure 3.1: LR-PON architecture with a connected cloudlet in each access zone.

(OLT). *Optical network units* (ONUs), in each access zone, can either be part of a wireless infrastructure or provide wired services for residential or business subscribers.

3.2 Cloudlet Placement

One of the main design considerations for integrating fog computing resources with optical access networks is the optimal placement of the fog cloudlets. This has recently become an active area of research following similar studies on placing cloudlets within wireless network infrastructures [83]–[85]. The challenge lies in making the computing resources as close as possible to end devices without requiring expensive architectural modifications or excessive fiber deployments. Common choices for cloudlet placements in optical access networks are near the CO, at the *remote nodes* (RNs), or in the field close to the ONUs. Placements may also be a hybrid of these three possibilities, as was proposed in [67].

Although placing a common fog cloudlet at the CO may be the most economical option in terms of the number of deployments, it may not be the best in terms of latency. Cloudlets at RN or

ONU locations may offer much lower latencies but would entail a relatively larger number of deployments. An RN-cloudlet will have the advantage of being connected to all the ONUs that branch out from that RN through their preexisting fiber links. However, this may require optical conversions as well as packet filtering and switching to take place at the RN. On the other hand, ONU-cloudlets are more dedicated and offer the least latency but can be the most expensive due to the large numbers of ONUs, especially in large-scale networks and dense deployment scenarios. Each ONU would also require a dedicated link to connect it to its cloudlet as well as a corresponding transceiver. In this chapter, we propose connecting a single powerful cloudlet, in each access zone, to the ONU, or subset of ONUs, serving the largest number of constrained devices. Other ONUs may either form their own virtual cloudlets through the aggregation of their connected devices having significant or available resources (e.g., an enterprise building with a large number of fixed PCs) or they can offload their data to another ONU with a connected cloudlet.

3.3 Cloudlet Offloading

Another key consideration for fog integration is offloading to the cloudlet. The offloading considered in this work is the optical offloading, which is the forwarding of offloaded traffic, once it reaches an ONU, to the closest cloudlet. In other words, the first offloading hop from the resource-constrained device to the parent ONU is not considered in this work since it has been studied in many other works (usually in a wireless setting).

While ONU-cloudlets have their own dedicated links, CO and RN-cloudlets may be shared between multiple ONUs through the existing fiber links that are primarily used for communicating with the OLT. The ONUs may then either use an additional time-shared wavelength to communicate with their common cloudlet or use a portion of their existing upstream bandwidth for OLT communications, as illustrated in Figure 3.2. The former approach would require placing

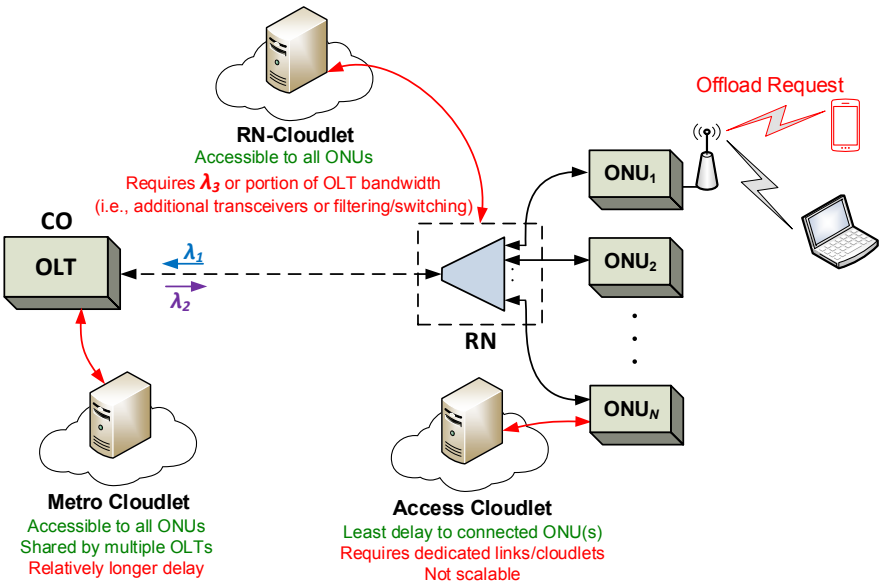


Figure 3.2: The offloading problem in a typical PON.

additional ONU transceivers and dedicating part of the optical spectrum for cloudlet communications, whereas the latter may greatly be affected by the underlying bandwidth allocation scheme and may also affect regular upstream traffic. In case of an RN-cloudlet, the latter approach may additionally require optical-electrical conversions within the RN before packets can be switched either to the cloudlet (offloaded traffic), to the OLT (upstream traffic), or back to ONUs (inter-ONU traffic). Such modification to the nearly-passive architecture of LR-PONs may be necessary for the requirements of next-generation networks.

In the rest of this chapter, we study the offloading performance when using centralized and decentralized dynamic bandwidth allocation after tailoring them to carry both upstream and offloaded traffic. The main cloudlet placement we consider is when a powerful cloudlet is connected to only one ONU via a dedicated link in each access zone (i.e., access cloudlet). This is believed to be a highly practical architecture as well as a worst-case scenario for other ONUs with no directly-connected cloudlet since offloading from one ONU to another directly connected to the cloudlet poses a challenge in a typical PON. Using this architecture, we closely examine imposed

delays on the offloading process and develop an analytical model that reflects the major factors affecting its performance. We later consider alternative placements for the cloudlet (at the RN or the CO) to examine which placement would be optimal for each bandwidth allocation paradigm.

3.3.1 Centralized-Based Offloading

In centralized-based bandwidth allocation, the OLT arbitrates bandwidth allocation among the ONUs using GATE and REPORT messages, as was discussed in Chapter 2. By adapting the transmission window sizes of ONUs in accordance with their bandwidth demands conveyed in their cyclic reports, the bandwidth allocation becomes dynamic. *Service level agreements* (SLAs) of end-users can then be enforced by setting an upper bound to the allocated bandwidth (i.e., the window size) of each ONU. The granted transmission window for ONU_i, in centralized *dynamic bandwidth allocation* (DBA), then becomes:

$$W_i = \begin{cases} R_i, & if\ R_i < W_{i,\max} \\ W_{i,\max}, & if\ R_i \geq W_{i,\max} \end{cases} \tag{3.1}$$

which can alternatively be expressed as:

$$W_i = \min(R_i, W_{i,\max}) \tag{3.2}$$

where R_i is the reported queue size, constituting the requested transmission window by ONU_i, and $W_{i,\max}$ is its maximum allowed window size according to its SLA agreement. The dynamic polling cycle duration for N ONUs can then be expressed as:

$$C = \sum_{i=1}^{N} W_i + NT_g \leq C_{\max} \tag{3.3}$$

where T_g is the guard interval between successive ONU transmissions and $C_{\max}$ is the maximum cycle duration when each ONU is granted its maximum window. Because polling forms the basis of centralized DBA, the performance of centralized bandwidth allocation greatly depends on the

round-trip time (RTT) imposed on the bandwidth negotiation messages between the OLT and ONUs. Due to the long network reaches of LR-PONs, the corresponding long RTTs greatly affect the packet delays causing the centralized allocation performance to ultimately degrade [82].

In order to offload data from one ONU to another using centralized polling, in an EPON, a few proposed modifications need to be made. First, the ONU's REPORT message needs to report data that is to be offloaded $R_{i,off}$ in addition to regular upstream traffic. Although offloaded traffic can alternatively be treated as regular traffic, in a transparent approach, distinctively reporting offloaded traffic will allow the OLT to give different priorities to each traffic type and thus have more control over the differentiation of services. This is easily achievable since the MPCP REPORT message has unused space reserved for future use. Reporting offloaded traffic is also essential for service request models discussed in following chapters, where the cloudlet has to accept the request before the data can be offloaded. The request must then specify the size of the data, the type of the service, and its delay constraints.

After receiving the ONU's REPORT, the OLT may either grant the ONU a transmission window that fits both types of data, if the ONU has enough unused bandwidth (i.e. $W_{i,max} - R_i \geq R_{i,off}$), or divide the transmission window between the two types of data giving one more preference over the other. Offloaded data must then be segmented into Ethernet frames that include their destination address. After the ONU offloads the data, the OLT would then forward it to the destination ONU (that is connected to the cloudlet), which imposes an additional propagation delay, as illustrated in Figure 3.3. A more convenient architecture for centralized allocation would therefore be placing an auxiliary cloudlet at the CO. This would alleviate offloading delays of ONUs with no directly-connected cloudlet. Of course, using just the CO-cloudlet will elongate the offloading delays of the ONU or set of ONUs that were to be directly connected to an ONU-cloudlet and therefore affect the services offered to their connected devices.

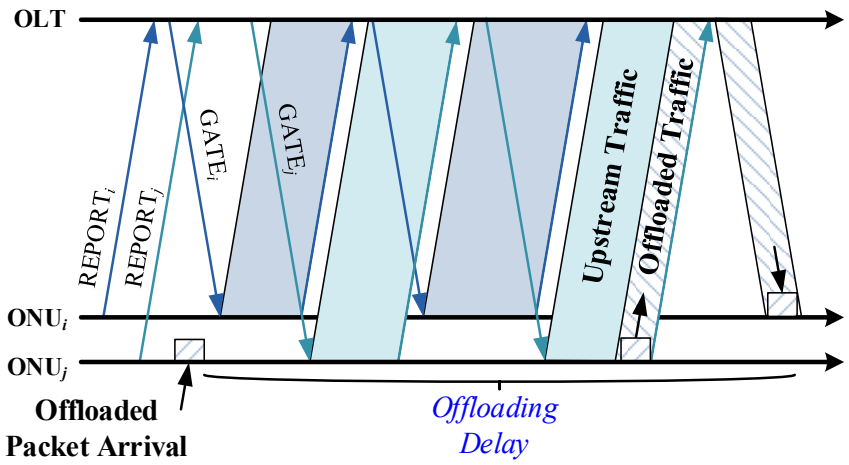


Figure 3.3: Centralized offloading delay to an indirectly-connected ONU cloudlet.

3.3.2 Decentralized-Based Offloading

As mentioned in Chapter 2, decentralized bandwidth allocation was introduced as a way of dealing with the degradation in centralized bandwidth allocation due to the long propagation delays of LR-PONs. In such schemes, the ONUs themselves manage their upstream media access instead of having to report their buffer status to the remote OLT and then wait for a grant to transmit. This allows pre-transmission delays to become independent of the ONUs' distances from the OLT. The concept of decentralized allocation is therefore somewhat analogous to that of moving the computing towards the network edge to improve the computing latency.

The main challenge facing decentralized allocation is acquiring direct communications among ONUs. For that, a decentralized scheme was introduced in Chapter 2, in which communication between ONUs was achieved by selectively reflecting back a specific wavelength facilitating an *out-of-band* (OOB) multipoint-to-multipoint channel between the ONUs [29]. The scheme was called *Taking Turns with Adaptive Cycle Time* (TTACT) since it allowed the ONUs to take turns transmitting on the upstream channel by communicating together on the additional channel, which requires additional transceivers to be placed within ONUs. An alternative

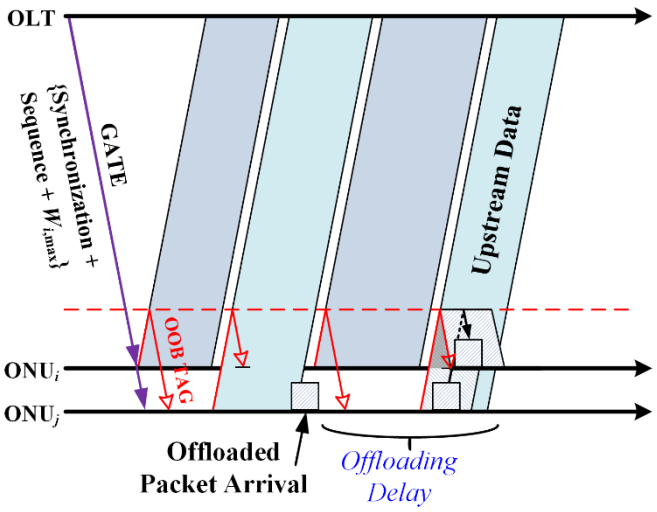


Figure 3.4: Decentralized out-of-band offloading delay.

approach for inter-ONU communications, also proposed in [29], was to place a MAC-layer switch within the RN and use in-band control messages that are to be filtered and broadcasted back to the ONUs. Such modification might be more preferable to some network operators rather than using additional ONU transceivers. It would also be very suitable for an RN-cloudlet configuration that already has active packet filtering within the RN.

Although the OOB channel in TTACT was only used to exchange media access tag frames between ONUs, in this work we propose to additionally use it to offload traffic from one ONU to another. This would speed up the offloading process in contrast to having to send the data first to the OLT, as would be done in a typical centralized scenario. Such configuration is illustrated in Figure 3.4, where a packet (received from a connected device to ONU$_j$) is offloaded from ONU$_j$ to ONU$_i$, on the OOB channel, when it is ONU$_j$'s turn to transmit. The available OOB offloading window (parallel to the in-band window) for any ONU$_i$ can be calculated as:

$$W_{i,OOB} = \left(\frac{W_i}{R_{up}} - D_{i,i+1}^{prop} \right) R_{OOB} - L_{tag} \tag{3.4}$$

where $D_{i,i+1}^{prop}$ is the propagation delay between ONU$_i$ and the following ONU in the transmission

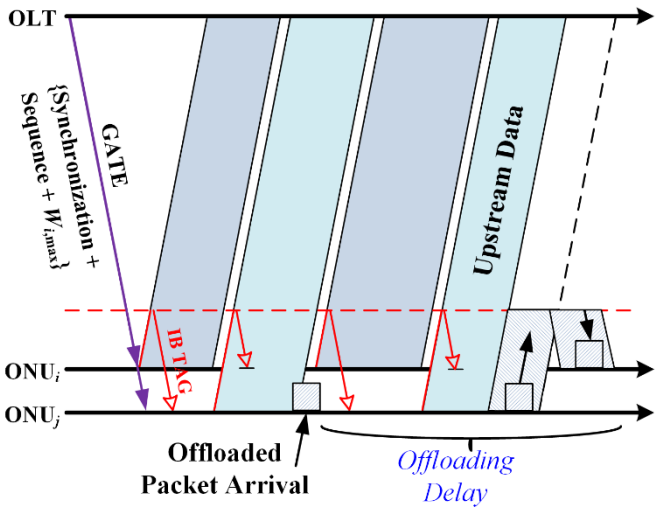


Figure 3.5: Decentralized in-band offloading delay.

sequence, R_{up} and R_{OOB} are the rates of the upstream and control wavelengths, respectively, and L_{tag} is the length of the tag frame. Accounting for propagation delays between ONU$_i$ and ONU$_{i+1}$ ensures that the tag message of the next ONU is not affected or delayed by this OOB offloading. Because W_i is the only variable in Eq. (3.4), an ONU that is about to offload data may choose to reserve its maximum window ($W_i = W_{i,\max}$) on the upstream, even if it will not fully utilize it, to give itself the maximum OOB window ($W_{i,OOB}$) for offloading.

Alternatively, in case of the in-band control scheme mentioned earlier, where a forwarding device within the RN filters inter-ONU communication frames and sends them back to the ONUs, an offloading ONU may only use the unutilized upstream bandwidth portion in its transmission window ($W_{i,\max} - R_i$) to offload data. This in-band offloading scenario is illustrated in Figure 3.5 and is similar to what is done in the centralized case. Here, however, the time it takes to send back offloaded data on the downstream will not be as long since forwarding is carried out in the RN rather than the OLT. A more convenient architecture for in-band decentralized allocation would therefore be to place an auxiliary cloudlet at the RN. This would alleviate offloading delays of ONUs with no directly connected cloudlets.

3.4 Offloading Performance Analysis

In this section, we take a closer look at the offloading process by analyzing its delay components within each allocation paradigm through a developed analytical framework.

3.4.1 Centralized-Based Offloading

Because ONUs have no direct link between them in the centralized case, offloading delays become highly dependent on upstream bandwidth allocation delays, which are in turn affected by the long RTTs between the OLT and ONUs. Once an ONU receives offloaded traffic from a connected device, it will first need to wait for a certain period of time before it can send its REPORT to the OLT, requesting bandwidth for this newly received traffic. This delay is dependent on its place in the polling cycle, which, on average, would be:

$$T_{report} \approx C_{avg} / 2 \tag{3.5}$$

Assuming that the OLT does not deprioritize upstream traffic (i.e., it only allows offloaded traffic to be transmitted using unused upstream bandwidth), it will start granting the ONU upstream bandwidth up to its $W_{i,\max}$ until it finishes offloading. This would take N_{off} cycles imposing another delay of approximately:

$$C_{avg} + N_{off}(C_{avg} + W_{i,\max} - R_{i,avg}) \tag{3.6}$$

where $R_{i,\text{avg}}$ here reflects the average amount of upstream traffic in the next N_{off} cycles and $(W_{i,\max} - R_{i,\text{avg}})$ represents the average unutilized bandwidth that can be used by ONU_i to transmit the offloaded data. N_{off} can therefore be expressed as:

$$N_{off} = \frac{L_{off} / R_{up}}{W_{i,\max} - R_{avg}} \tag{3.7}$$

where L_{off} is the total size of offloaded data including overheads. After receiving the data, the OLT will send the data back on the downstream to the ONU that is connected to the cloudlet. This takes:

$$D_{i,j}^{prop} + D_{OLT}^{process} + L_{off} / R_{down} \tag{3.8}$$

where $D_{i,j}^{prop}$ is the propagation delay between ONU$_i$ and ONU$_j$, which is $D_{i,OLT}^{prop} + D_{j,OLT}^{prop} \approx 1RTT$. The total centralized offloading delay is then:

$$D_{off} \approx \frac{3C_{avg}}{2} + N_{off}C_{avg} + \frac{L_{off}}{R_{up}} + \frac{L_{off}}{R_{down}} + D_{OLT}^{process} + D_{i,j}^{prop} \tag{3.9}$$

If the offloaded data required a response or some result to be sent back (e.g., if offloaded data constitutes input data for a computational task), the data will first need to be processed at the cloudlet, before sending back the result, which would undergo a retrieval delay that is similar to the delay in Eq. (3.9):

$$D_{ret} \approx \frac{3C_{avg}}{2} + N_{ret}C_{avg} + \frac{L_{ret}}{R_{up}} + \frac{L_{ret}}{R_{down}} + D_{OLT}^{process} + D_{j,i}^{prop} \tag{3.10}$$

where L_{ret} is the size of the cloudlet response and N_{ret} is the number of cycles it takes to transmit this response to the OLT using the available bandwidth of ONU$_j$.

As mentioned before, an alternative cloudlet placement here would be near the OLT. That way, offloaded data will not need to be transmitted from the OLT all the way back to the ONU-cloudlet. The propagation delays in Eqs. (3.9) and (3.10) will then be cut by half. However, using only this cloudlet will make it far from all ONUs and their connected devices, including the large number of constrained devices that would have been just a single hop away from an ONU-cloudlet. The offloading delay for these devices would then increase by:

$$\frac{3C_{avg}}{2} + N_{off}C_{avg} + \frac{L_{off}}{R_{up}} + D_{i,OLT}^{prop} \tag{3.11}$$

It is therefore better to make such OLT-cloudlet an auxiliary cloudlet that serves the ONUs that are not directly connected to the ONU cloudlet.

Looking at the factors affecting the offloading delays in Eqs. (3.9) and (3.10), it is clear that they are mainly affected by two factors. The first is the number of offloading cycles N_{off}, which depends on the size of the offloaded data, the upstream data rate, and the available bandwidth, as can be seen from Eq. (3.7). Huge data sizes and heavy network loads can both lead the number of cycles to considerably grow making N_{off} the dominant factor in centralized offloading delays. The second main factor is the propagation delay, which depends on the cloudlet placement. This factor, however, will only become significant when the number of offloading cycles is too low. For instance, if the size of the data is too small compared to the transmission rate, the data may require very few offloading cycles making the propagation delay a dominant factor.

3.4.2 Decentralized-Based Offloading

In the decentralized scenario, an ONU can access the common upstream channel after a turn delay that is on average:

$$T_{turn} \approx C_{avg} / 2 \tag{3.12}$$

ONU_i can then use the OOB window in Eq. (3.4) to directly transfer offloaded data to ONU_j. If ONU_i chooses to reserve its maximum upstream window to gain more OOB channel time, the offloading transmission will approximately take:

$$\frac{L_{off} / R_{OOB}}{W_{i,OOB}} (C_{avg} + W_{i,\max} - R_{i,avg}) + D_{i,j}^{prop} \tag{3.13}$$

where the propagation delay here is different from the centralized case since OOB transmissions are reflected at the RN. The total offloading delay can then be expressed as:

$$D_{off} \approx \frac{C_{avg}}{2} + \frac{L_{off} / R_{OOB}}{W_{i,OOB}} (C_{avg} + W_{i,\max} - R_{i,avg}) + D_{i,j}^{prop} \tag{3.14}$$

where N_{off} can be expressed as:

$$N_{off} = \frac{L_{off} / R_{OOB}}{W_{i,OOB}} = \frac{L_{off} / R_{OOB}}{\left(W_{i,\max} / R_{up} - D_{i,i+1}^{prop}\right) R_{OOB} - L_{tag}} \tag{3.15}$$

Similarly, the retrieval delay will be:

$$D_{ret} \approx \frac{C_{avg}}{2} + \frac{L_{ret} / R_{OOB}}{W_{j,OOB}} (C_{avg} + W_{j,\max} - R_{j,avg}) + D_{j,i}^{prop} \tag{3.16}$$

On the other hand, if the in-band decentralized configuration is used (Figure 3.5), the offloading delay can be then expressed as:

$$D_{off} \approx \frac{C_{avg}}{2} + N_{off} C_{avg} + \frac{L_{off}}{R_{up}} + \frac{L_{off}}{R_{down}} + D_{i,j}^{prop} \tag{3.17}$$

where N_{off} here can be expressed with Eq. (3.7). The retrieval delay would then be:

$$D_{ret} \approx \frac{C_{avg}}{2} + N_{ret} C_{avg} + \frac{L_{ret}}{R_{up}} + \frac{L_{ret}}{R_{down}} + D_{j,i}^{prop} \tag{3.18}$$

An alternative cloudlet placement here would be at the RN. This will reduce the offloading delays for all other ONUs making its offloading delays:

$$D_{off} \approx \frac{C_{avg}}{2} + N_{off} C_{avg} + \frac{L_{off}}{R_{up}} + D_{i,RN}^{prop} \tag{3.19}$$

where the propagation delay here would only be to the RN.

Looking at Eqs. (3.14) and (3.17), it is clear that decentralized-based offloading is affected by the same two factors affecting centralized offloading delays. The first is the number of offloading cycles N_{off}, which again depends on the size of the offloaded data, the data transmission rate, and the available bandwidth. However, the transmission rate here may either be the OOB

transmission rate or that of the upstream, depending on whether the data is offloaded out-of-band or in-band. The second is the propagation delay, which is minimal for RN and ONU cloudlets. The first factor is therefore expected to be the dominant one for decentralized offloading delays.

3.5 Numerical Results

For our simulation, we consider a 100km LR-PON consisting of an OLT and 16 ONUs. The RN is located 95km away from the CO and the ONUs are placed randomly in a 5km access zone. We consider both asymmetric and symmetric 10G-EPON configuration (IEEE 802.3ah/v), where ONUs share an upstream wavelength of either 1Gbps or 10Gbps. Devices from the access side have an access rate of either 100Mbps or 1Gbps making the system throughput less than the peak-aggregated load from all ONUs. Each ONU is assumed to have a buffer of 10MB and the traffic model used is self-similar Ethernet traffic constructed from alternating Pareto-distributed streams with a Hurst parameter of 0.8, similar to the model used in [18], [29], and [82]. To compare between the different allocation schemes, $C_{\max}$ is set to 5ms for all schemes, which is found to be optimal for a 100km span. The inter-transmission guard interval T_g is set to 5µs for both in-band and out-of-band traffic. For the out-of-band decentralized scheme, we choose R_{OOB} to match the used upstream rate since it will be used to carry offloading traffic.

3.5.1 General Offloading Performance with a 1Gbps Upstream

Figure 3.6 illustrates the average offloading delays for all schemes, under different network loads, when 5MB of data is offloaded from an ONU to another connected to a cloudlet. The figure shows how analytical delays computed using Eqs. (3.9), (3.14), and (3.17) are very close to those obtained from simulation. Figure 3.7 presents the relative error ($\left|D^*_{off} - D_{off}\right| / D^*_{off}$) between

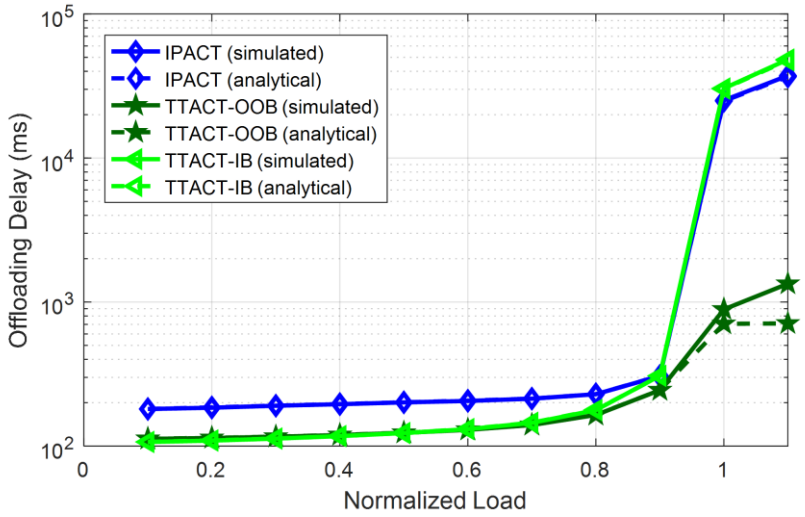


Figure 3.6: Analytical and simulated offloading delays.

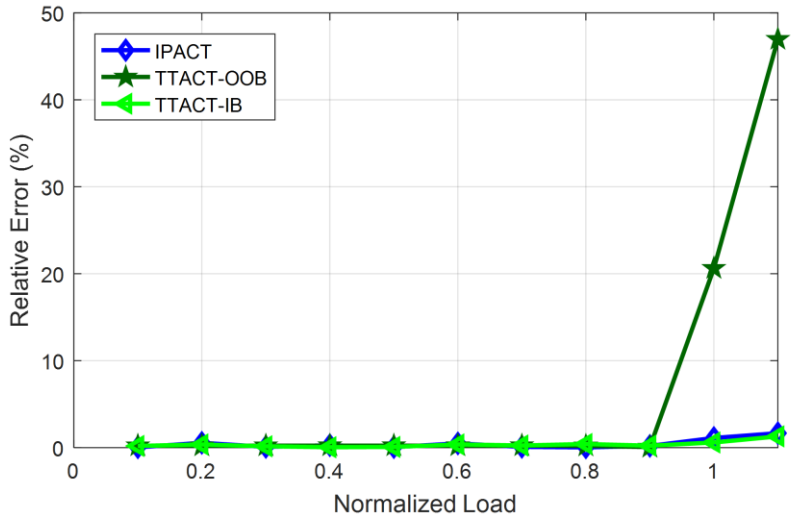


Figure 3.7: Relative error of offloading delays.

offloading delays obtained from simulations (D_{off}^{*}) and the analytically computed ones (D_{off}), where it can be seen that the approximation errors are usually below 1% under normal loads. Under normal network conditions, both decentralized schemes achieve slightly lower offloading delays compared to centralized polling. Although centralized offloading delays are greatly affected by the network span (since offloaded data must always be sent back to the ONU-cloudlet after first

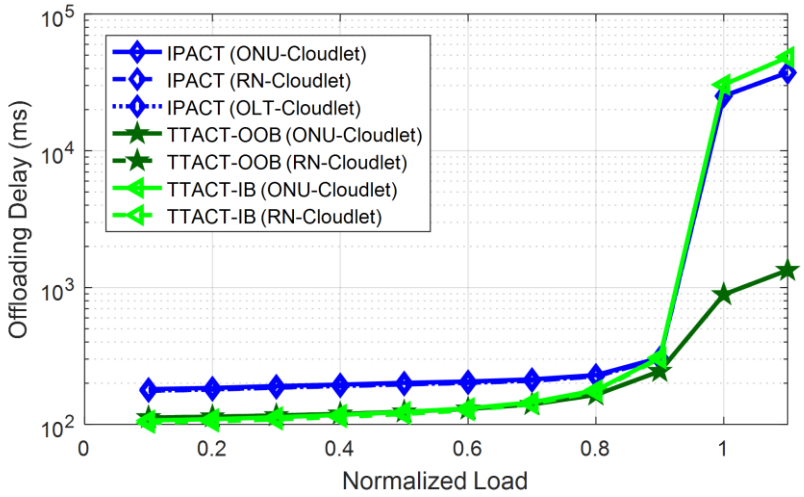


Figure 3.8: Comparison of offloading delays with alternative cloudlet placements.

being received at the OLT), the network span is not found to be the dominant factor for offloading delays in Figure 3.6. This is because the RTT in a 100km LR-PON is ~1ms, whereas the offloading delay under normal loads is shown to be above 100ms making the network span only responsible for ~1% of the 5MB task delays.

3.5.2 Performance with Alternative Cloudlet Placements

Figure 3.8 illustrates the change in offloading performance with different cloudlet placements. Even though placing the cloudlet at the OLT reduces propagation delays of centralized-based offloading by half, it still does not show significant improvement. This further proves that the propagation delay is not the dominating factor in centralized-based offloading and that the offloading bandwidth is, in fact, the true bottleneck. The same effect is observed for decentralized-based offloading, where each ONU is limited by the maximum upstream transmission window even when using the additional channel for offloading.

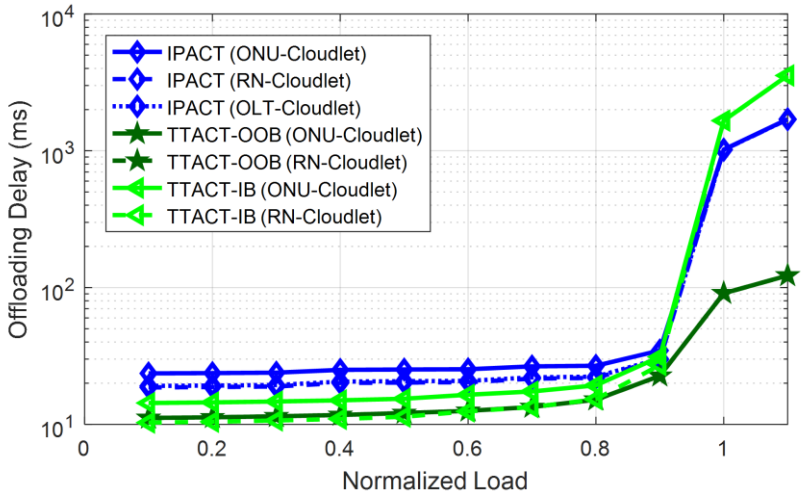


Figure 3.9: Alternative cloudlet placements with a 10Gbps upstream.

3.5.3 General Offloading Performance with a 10Gbps Upstream

Figure 3.9 illustrates the offloading performance again with different cloudlet placements for a symmetric 10Gbps EPON. Increasing the upstream rate to 10Gbps reduces the number of offloading cycles and corresponding offloading delays by an order of magnitude. This increase in transmission rate leads propagation delays to become the dominating factor. The alternative cloudlet placements therefore have significant effects on the delays with high transmission rates. While placing the cloudlet at the OLT leads to reducing centralized-based offloading delays by nearly half, the improvement in the in-band decentralized scheme due to using an RN cloudlet makes its performance almost similar to that of the OOB scheme under normal network conditions.

3.5.4 Effect of Offloading on Upstream Delays

In our simulation, offloaded traffic is given no priority over regular upstream traffic and is only transmitted with excess bandwidth. However, offloading still affects upstream packet delays, as illustrated in Figure 3.10. The figure shows the corresponding increase in upstream packet

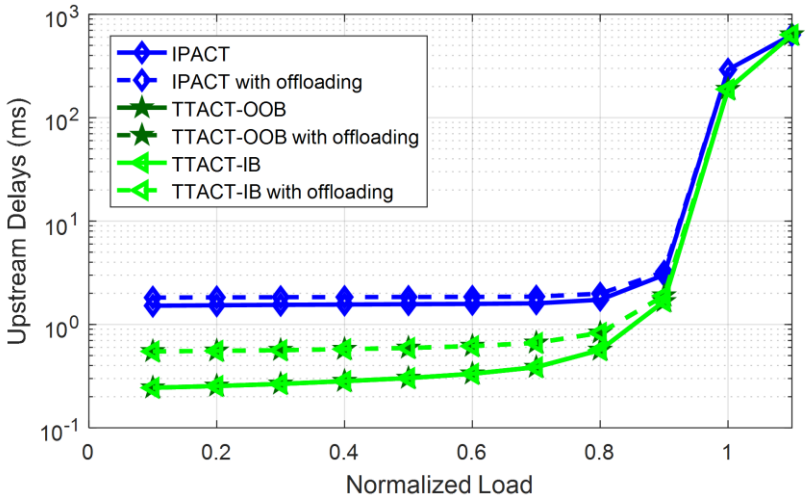
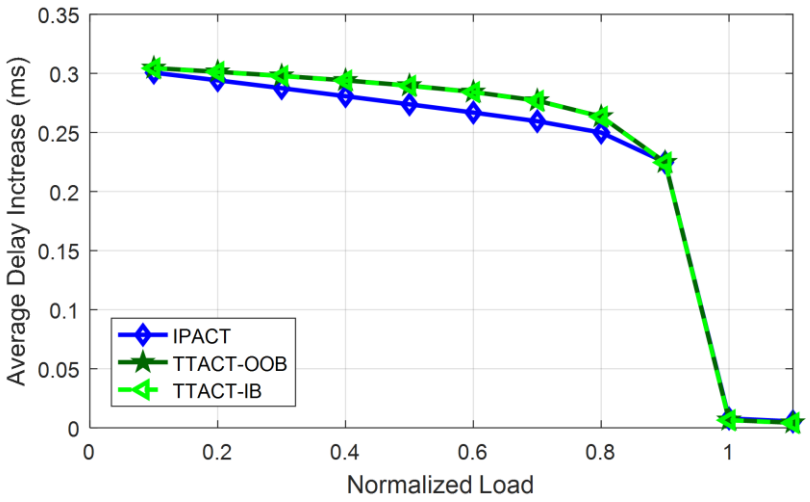


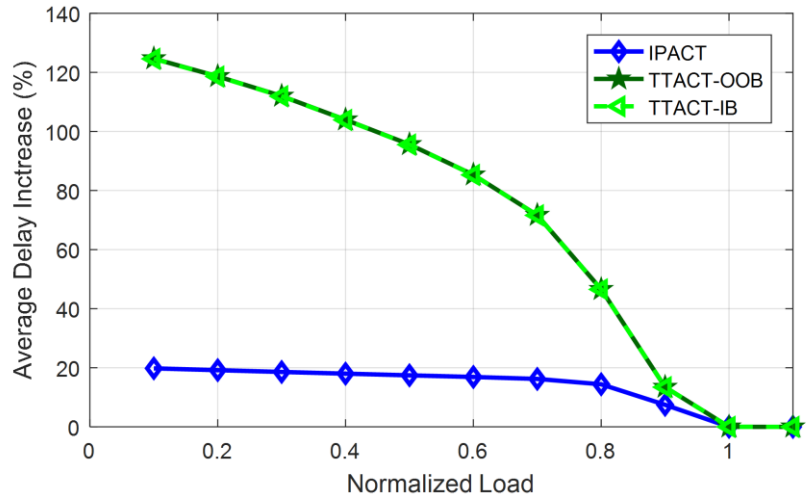
Figure 3.10: Effect of offloading on upstream delays.

delays, under light loads, due to the elongation of the upstream cycle during an offloading phase. Under heavy loads, there is rarely any excess bandwidth for offloading and the cycle is already too long due to upstream traffic. Even though offloading is done out-of-band in TTACT-OOB, upstream delays are still affected since ONUs are set to reserve their maximum transmission windows and give themselves the maximum possible OOB window for offloading. While this speeds up offloading, it delays queued upstream traffic of other ONUs by the corresponding cycle elongation ($W_{i,\max} - R_{i,avg}$), as can be seen in Eq. (3.14). A similar elongation takes place for in-band TTACT, in which the data is offloaded on the upstream channel. However, it is worth noting that, in all allocation schemes, this effect is temporary and only lasts while offloading or retrieving results takes place. In other words, the effect only lasts for N_{off} and N_{ret} transmission cycles.

Figure 3.11a–b better illustrates the effect on upstream traffic delays by only showing the increase in delays. While upstream delays are shown to increase by around 0.3ms for all schemes under normally loaded conditions, this increase constitutes a higher percentage in the decentralized



a) Absolute delay increase



b) Delay increase percentage

Figure 3.11: Increase in upstream delays due to offloading.

schemes. Still, the effects shown are for a single ongoing offload. In case of multiple concurrent offloads, the upstream cycle will elongate further resulting in more increase in upstream delays.

3.5.5 Effect of Offloaded Data Size on Offloading Delays

The offloading delay is proportionate to the size of the data being offloaded (L_{off}), as can be seen in Eqs. (3.9), (3.14), and (3.17). Increasing the size of offloaded data, however, exposes a

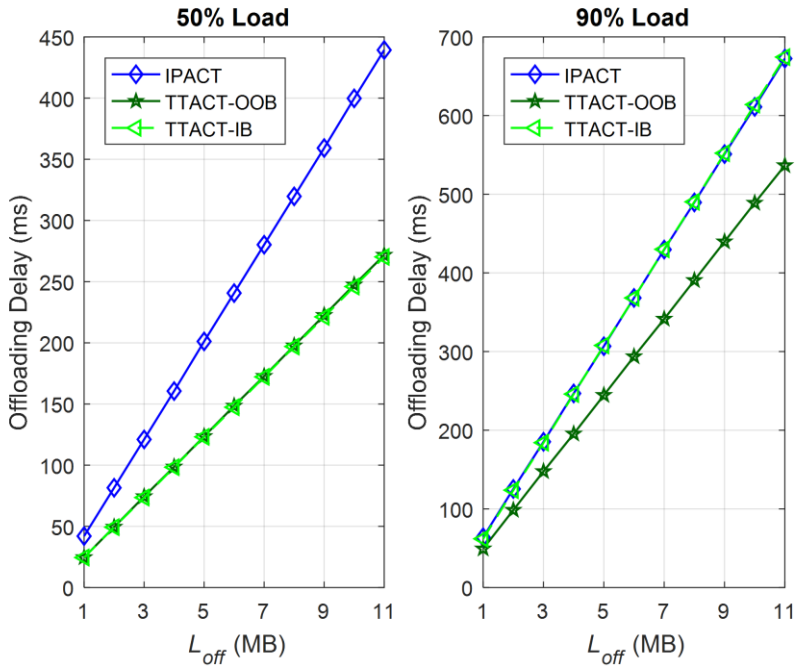


Figure 3.12: Effect of offloaded data size on offloading delays.

different level of degradation in each scheme according to its elemental delay components. This is illustrated in Figure 3.12 under 50% and 90% network loads, where it can be seen how the degradation in centralize-based offloading becomes more severe than that of decentralized-based offloading. While both decentralized schemes are found to have very similar performances under normal loads, the in-band scheme has worse performance under heavy loads that is almost similar to centralized allocation. This is because in-band offloading is done using excess upstream bandwidth, which becomes very limited under heavy loads. On the other hand, even though out-of-band offloading should not be affected by upstream traffic, the elongation of the cycle due to upstream traffic, under heavy loads, results in a corresponding increase in offloading delays.

3.6 Conclusions

In this chapter, we shed some light on the design considerations of integrating fog computing with long-reach optical access networks from a bandwidth allocation perspective.

We modeled and simulated the offloading performance when data is offloaded from one ONU to another that is connected to an access cloudlet. We compared between centralized and decentralized-based offloading performances pointing out alternative cloudlet placements for each scheme and the associated tradeoffs. Modifying the network architecture to support decentralized allocation was shown to achieve significant gains in the offloading performance. Not only does decentralized-based offloading experience lower delays, but its performance is also not degraded as much as centralized-based offloading when the size of the offloaded data increases.

Chapter 4

Towards Edge Computing in LR-PONs

In the previous chapter, we examined the integration of fog computing with optical access networks through employing cloudlets at the edge of the network. In this chapter, we explore the possibility of using the resources of the network itself. First, we formulate our edge computing problem before comparing the performances of centralized and decentralized-based service delays. We also propose novel modifications to the decentralized media access scheme to better support exchanging edge traffic. Service delays in each scheme are then analyzed through a developed analytical framework and compared against simulation results. The notations of this chapter are summarized in Table 4.1 along with their description and default values.

4.1 Edge Computing Problem Assumptions

As was mentioned in previous chapters, some studies believed that the most convenient way of placing storage and computing resources within an optical access network would be through connecting a cloudlet to each ONU [66]. However, because ONUs are the most numerous component in such networks, this may not be the most feasible nor cost-effective solution. A more scalable solution would be to connect a single powerful cloudlet to each *remote node* (RN), as was proposed in [63], which would allow the cloudlet to serve all connected ONUs in its access zone while still being relatively close to constrained devices. This, however, requires packet switching

Table 4.1 Symbol Definitions and Simulation Values

Symbol	Description	Value
$L_{\max}$	Maximum network span	100km
$L_{i,SC}$	ONU_i's distance from splitter/combiner	0–5km
L_{SC}	Splitter distance from OLT (feeder span)	95km
R_{down}	Downstream transmission rate	10Gbps
R_{up}	Upstream transmission rate	1Gbps
R_{OOB}	Out-of-band channel transmission rate	0.1–1Gbps
R_{user}	Connected users' upstream rate	100Mbps
N	Number of ONUs	16
Q	ONU buffer size	10MB
L	Normalized network load	0.1–1.1
$C_{\max}$	Max. cycle duration	5ms
T_g	Guard interval	5µs
T_M	Thread guard interval (multi-thread)	5µs
M	Number of threads (multi-thread)	3
X	Cycle extension factor (TTACT-UITX)	2,3,4
D	Size of offloaded data	5MB,10-110MB
S	Number of selected ONUs	5
E	Size of each computational result	$1.5D$

to be carried out at the RN to filter out cloudlet traffic from OLT traffic. In this chapter, we consider a third option, where the computations are proposed to be done by the ONUs themselves.

In [65], it was proposed that the PON itself may serve as a cloud by using the computational and storage resources of the OLT and numerous ONUs. Although the OLT typically has double the resources of a single ONU, the ONUs are much larger in number, such that, together, they hold most of the network's computing power. For instance, with the least PON split of 16, the computing and storage capabilities of the ONUs would be 8 times that of the OLT (see [65] for detailed PON capabilities). The ONUs are also in close proximity to connected devices as well as to each other. However, to make full use of the ONU resources, offloaded tasks may need to be distributed among them, which requires inter-ONU communications and management. Before examining which allocation paradigm would be a better fit to fully support such a cooperative setting, we first formulate our edge computing problem.

For our edge computation and service composition problem, we consider a LR-PON with N ONUs capable of accepting offloaded tasks from their resource-constrained connected devices. We then assume the following with regard to the available computational resources:

1. each ONU is capable of running some computational tasks besides its main functions as well as using part of its memory for storage purposes,
2. ONUs in the same network are identical (homogeneous computing resources),
3. the network has enough resources to carry out more than one offloaded task,
4. a task can be composed of services offered by an ONU or from those offered by its connected devices as long as they meet delay constraints and have acceptable battery levels.

We then assume the following for a given offloaded task:

1. a service task T is first requested by a client device connected to ONU_c,
2. if the service task T is accepted by ONU_c, it is to be carried out using some offloaded data and an associated set of operations that are R bytes in size,
3. R is fragmented into Ethernet packets that add up to D bytes in size including overheads,
4. task T may be composed of multiple services that can be broken into S subtasks, which may then be divided among S ONUs (or connected devices) that are offering these required services ($S \geq 1$, with 1 meaning that the task is undividable),
5. the output of each subtask may vary and is E_s bytes in size ($s \in S$),
6. either the OLT (in the centralized case) or the client's node ONU_c (in the decentralized case) will be responsible for composing the service or at least finding an entity with enough resources to manage the service composition,
7. the composition manager will be responsible for composing the service from local or remote resources, collecting the final result, and sending it back to ONU_c.

Finally, we assume that only unused bandwidth is used for exchanging edge traffic. This ensures, that regular upstream traffic will be virtually unaffected by edge traffic. Now, for offloading and task distribution/retrieval to take place, the mechanisms are different in each allocation paradigm as discussed in the following sections.

4.2 Offloading and Service Composition

Before a device can offload a task to the network, the device must first construct a requirement list that specifies the services required and the corresponding *quality of service* (QoS) constraints for each service. For instance, if the service is some required processing, the device must specify the amount of required computation as well as some QoS parameters, such as a delay constraint for that computation. The device then embeds this list into its service request message that is sent to the ONU to which it is connected (i.e., ONU_c). Once ONU_c receives this service request, four phases are required to compose the requested service:

Composition Manager Selection Phase:

The composition manager is responsible for composing the service, collecting the service result and sending it back to the origin where the service was requested. Because, in our setting, the service request originates from a device connected to ONU_c, we have one of three possibilities:

1. ONU_c has the required resources and offered services to fulfill the service request,
2. ONU_c becomes a composition manager that composes the service from other network entities that have the necessary resources,
3. ONU_c neither has the resources nor the functionality to become the composition manager and must elect another network entity to become the composition manager.

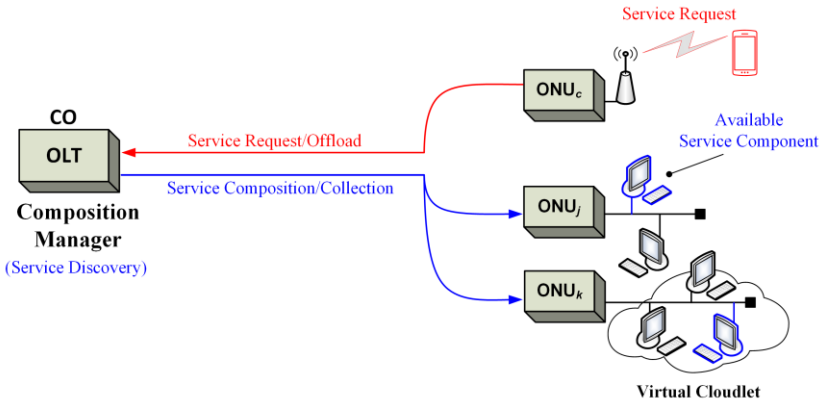
The first case offers no challenge since the service request will be answered by the

connected ONU with the least delay. In the second case, ONU_c becomes the composition manager and is responsible for allocating resources from the network to carry out the service. We therefore consider this case a decentralized-based service composition. In the final case, either ONU_c does not have the resources for composing the service or such function belongs to the OLT, which either becomes a composition manager or elects one. If the OLT becomes the composition manager, it is then a centralized-based service composition scenario. Otherwise, the composition manager would have to be elected from among the ONUs based on the availability of their resources and services offered. This may be done by checking their cyclic updates and selecting the ONU that currently has the highest available computational capacity, offers the requested services, and meets the QoS requirements of these services. Similar to [49], this ONU election may be based on a utility function U, which can be expressed as:

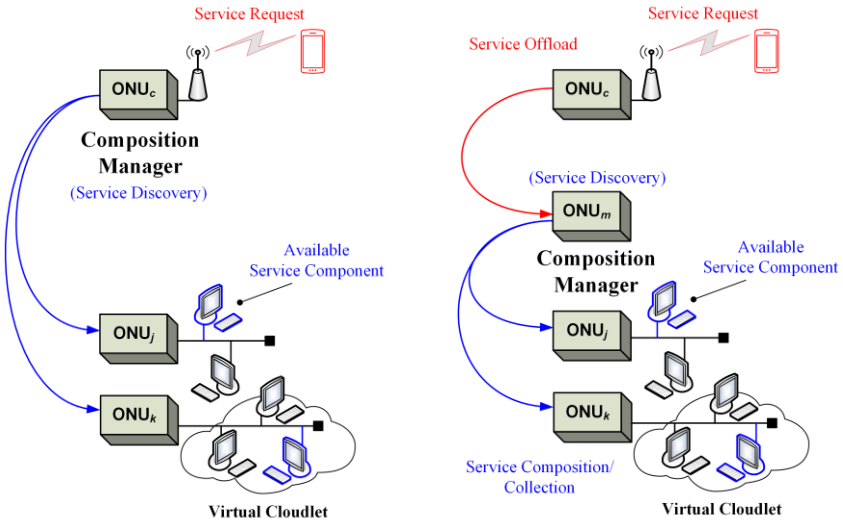
$$U(\mathrm{ONU}_i) = \alpha N(\mathrm{ONU}_i) + \beta M(\mathrm{ONU}_i) + \gamma Q(\mathrm{ONU}_i) \qquad (4.1)$$

where N and M are the numbers of services currently being offered by ONU_i itself and from the devices connected to it, respectively. Q is a metric that reflects the similarity between the services offered and those requested, whereas α, β, and γ are weights that may be used to prioritize one utility parameter over the others. After the composition manager ONU_m is successfully elected, the task has to be passed on to it. ONU_c can then accept the service request from its attached device, and forward offloaded service data to ONU_m.

In the remainder of this chapter, we consider the OLT to be the service manager in centralized-based bandwidth allocation since there are no direct inter-ONU communications in a typical centralized-based network. However, because ONUs are much larger in number and may be connected to even a larger number of devices, we assume that the OLT would then compose the service from their resources and offered services as well as those of their attached devices.



a) Service composition configuration in the centralized case.



b) Possible service composition configurations in the decentralized case.

Figure 4.1: Assumed service composition configurations.

On the other hand, we assume ONU_c to become the service manager in the decentralized bandwidth allocation case. The service is then composed of other nearby ONUs through inter-ONU communications. Alternatively, ONU_c may have to elect another ONU to become the service manager if it has no available resources to become the manager itself. These configurations are illustrated in Figure 4.1.

Service Discovery Phase:

In this phase, the selected composition manager performs the discovery process, in which it investigates all the services offered by nearby devices as well as by itself. The manager then selects the best available services that can be integrated together to provide the required service. The service discovery phase is usually composed of two main steps: forming a candidate list and ranking each candidate.

The first step requires the manager to have a list of the services currently being offered by devices in its vicinity along with their corresponding QoS metrics. This list may be already available to the manager through a locally cached description that is gathered from its connected devices through periodic updates. For example, the OLT in the centralized case may already know the services offered by each ONU from its previously received REPORT message. The services must then be represented by their computational or storage capabilities (e.g., 1GHz processing speed or 100MB memory storage availability).

The second step uses some information from the service request message to compute the rankings of the available services using a ranking function R, such as:

$$R(s_j) = w_c w_r sim(s_j, \boldsymbol{S_r}) \tag{4.2}$$

where w_c and w_r are weights reflecting the availability of a candidate service and the priority of a required service, respectively, whereas $sim(s_j, \boldsymbol{S_r})$ is a similarity function that matches between an offered service s_j and a set of requested services $\boldsymbol{S_r}$ of similar nature. If multiple instances of the same service exist, the weight w_r can be used to give preference to one over the others based on some metric. Examples of possible metrics may be the closest network entity in distance or the one having the least number of hops.

Service Integration and Execution Phase:

In this phase, the service manager coordinates the execution of the selected services in the order specified by the service request message. It also ensures the transfer of intermediate results, if any, from one service to another when necessary. Execution can therefore occur in a distributed manner, where partial results received from a service (executed on one device) are transferred to the following service (executed on another).

Result Collection Phase:

In this phase, any final generated output must be sent back to the device where the service request originated. This means that results must first be gathered by the manager (if they were computed using separate network entities) and then sent back to ONU_c.

In the following sections, we study how these service composition phases can be carried out in each bandwidth allocation scheme.

4.3 Centralized-Based Edge Computing

In this chapter, we use a different centralized allocation scheme for our study, namely multi-thread polling, which is believed by many studies to improve centralized allocation's performance in LR-PONs. In the following sections, we briefly introduce the basic principles of multithread polling, before turning our focus to its offloading performance in our edge-computing problem.

4.3.1 Multi-thread Polling

In previous chapters, it was explained how the OLT, in centralized bandwidth allocation, arbitrates the ONUs' bandwidth allocation based on their received reports. This allows the OLT to

adapt transmission window sizes of ONUs in accordance with their bandwidth demands, thus making the bandwidth allocation dynamic. With limited service, the OLT sets an upper bound to the window sizes of each ONU, making the granted transmission window for the i^{th} ONU:

$$W_i = \min(R_i, W_{i,\max}) \tag{4.3}$$

where R_i is the reported queue size and the requested transmission window for ONU_i, whereas $W_{i,\max}$ is the ONU's maximum allowed window size according to its *service level agreement* (SLA). The polling cycle duration for N ONUs can thus be expressed as:

$$C = \sum_{i=1}^{N} W_i + NT_g \leq C_{\max} \tag{4.4}$$

where T_g is the guard interval between successive ONU transmissions and $C_{\max}$ is the maximum cycle duration when each ONU is granted its maximum window. Figure 4.2 illustrates the different types of pre-transmission delays (i.e., delays imposed on a packet prior to its transmission), in such a centralized polling scheme, which are namely: *reporting*, *grant*, and *queuing* delays. It can be seen how the allocation performance greatly depends on the *round-trip times* (RTTs) imposed on the bandwidth negotiation messages exchanged between the OLT and ONUs. While this effect is not significant within traditional PONs with 10-20km spans, the RTTs become more severe when the span is further extended, causing the performance to ultimately degrade. One proposed way to deal with this degradation was to use multi-thread polling [18].

Figure 4.3 illustrates multi-thread polling and its associated pre-transmission delays. As can be seen from the figure, this polling technique is based on creating multiple interleaved parallel threads of communications between the ONUs and the OLT, each with its own report and grant messages. The cycle duration with M threads can then be expressed as:

$$C = \sum_{i=1}^{M} T_i + MT_M \leq C_{\max} \tag{4.5}$$

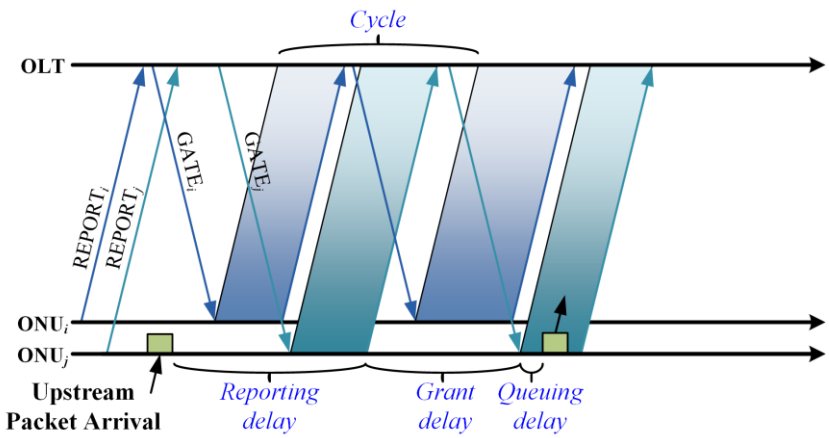


Figure 4.2: Interleaved polling and its pre-transmission delays.

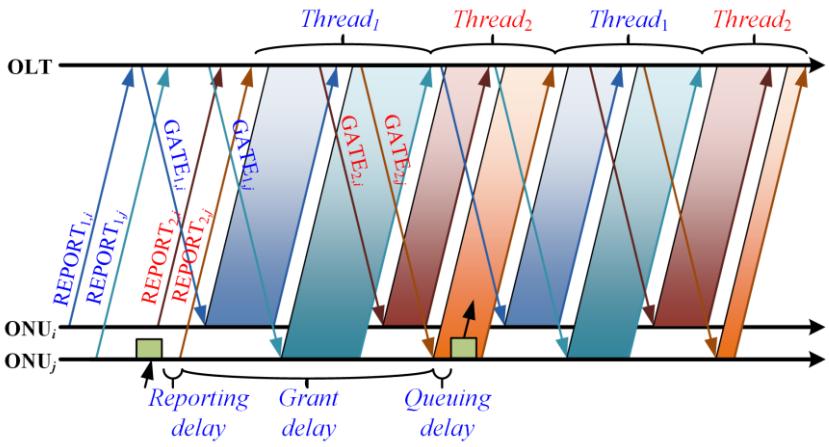


Figure 4.3: Multi-thread polling and its pre-transmission delays.

where T_i is the instantaneous duration of the i^{th} thread and T_M is the guard interval between adjacent threads. Splitting the cycle duration into multiple threads in such a way has been shown to reduce both reporting and queuing delays by a factor of M [18]. This is because using more threads entails using more reports, which leads to a packet waiting less time before being reported. Each thread will also have less data to transmit in each thread, which results in lower queuing delays. However, grant delays still remain highly dependent on the RTT.

Since it was first introduced, multi-thread polling has gained much popularity in the literature and opened the door for many proposed modifications and extensions (see for example

[27], [82], [86], [87]). Still, the original algorithm involves many mechanisms such as thread tuning and excess bandwidth distribution [18]. In this chapter, we consider it as the centralized candidate for supporting edge computing. To the best of our knowledge, multi-thread polling has never been studied for supporting edge computing.

4.3.2 Edge Computing with Multi-thread Polling

Similar to the model used in the previous chapter, ONUs within centralized allocation are assumed to only use excess bandwidth for offloading. With no direct inter-ONU communications in centralized allocation, the client's ONU (i.e., ONU_c) would first need to forward its service request to the OLT, which either becomes the composition manager or becomes responsible for selecting one from the ONUs, based on information gathered from its most recent reports. Since the allocation here is centralized, we assume that the service composition would also be centralized, where the OLT itself becomes the composition manager. The OLT will then check if the task can be broken into subtasks and check available ONU resources. This, of course, requires ONUs to continuously append the availability of their resources and offered services to their cyclic reports.

After allocating resources, the OLT will accept the service request and start granting ONU_c more upstream bandwidth up to its allowable maximum ($W_{c,\max}$). Additionally, in multi-thread polling, the OLT calculates unused bandwidth from normally-loaded ONUs and distributes it proportionately among heavily-loaded ones in each cycle [18]. A portion of this excess bandwidth may then be used by ONU_c to offload its service data to the OLT. Offloading however can take more than one cycle depending on the ONU's current upstream load (which affects its excess bandwidth), the network load (which affects its portion of excess bandwidth), and the size of the offloaded data. Once service data has been offloaded to the OLT, it will start forwarding it to the selected ONU or subset of ONUs (if the task is to be subdivided among S ONUs). The OLT may

Algorithm 1 Centralized algorithm executed by OLT

if (*REPORT* is received) **then**
  Determine ONU's next slot time and duration:
  R_i, $R_{i,off} \leftarrow$ extract ONU_i buffer status
  $W_i = \max(R_i + R_{i,off}, W_{i,\max})$
  if $(R_i + R_{i,off} > W_{i,\max})$ **then**
    mark ONU_i as a heavily-loaded ONU
    allocate ONU_i excess bandwidth at end of the thread
  end if
  $T_{i,start} = \max(T_{i-1,start} + W_{i-1} + T_g - RTT_i, CLK_{OLT})$
  Generate $GRANT_i$ frame
  $GRANT_i \leftarrow$ append $\{T_{i,start}, W_i, CLK_{OLT}\}$
  $T_{i,grant} = T_{i,start} - RTT_i$
  Edge computing module:
  C_i, $L_i \leftarrow$ extract computational availability and offered services
  $F_i \leftarrow$ extract if edge data is being transmitted
  $A_i \leftarrow$ extract if upcoming edge data is service request or output
  if ($F_i = 1$ & offloaded data has been fully received) **then**
    if (offloaded data is a result) **then**
      immediately forward to origin ONU
    else if (offloaded data is a service request) **then**
      $S \leftarrow$ check if task is dividable
      $\boldsymbol{A} \leftarrow$ check last $N - 1$ availability flags
      $\boldsymbol{S} \leftarrow$ target S ONUs from availability set $\boldsymbol{A}$
    end if
  end if
  if ($ONU_i \in \boldsymbol{S}$ & ONU_i has available resources) **then**
    assign task/subtask
    $GRANT_i \leftarrow$ append $\{task\_ID, subtask\_ID, \text{size}(D)\}$
  else if ($ONU_i \in \boldsymbol{S}$ & ONU_i has not enough resources) **then**
    remove ONU_i from set $\boldsymbol{S}$
    find replacement from set $\boldsymbol{A}$
  end if
  Send $GRANT_i$:
  if $(T_{i,grant} = CLK_{OLT})$ **then**
    send $GRANT_i$
  end if
end if

either send each ONU its corresponding data or alternatively broadcast all the data to the selected ONUs since the downstream forms a point-to-multipoint topology. After each ONU receives its subtask and input data, the OLT waits for the ONUs to finish their computations and upload their outputs using their excess bandwidth before sending back the results to ONU_c.

Algorithm 1 shows the proposed process that is to be executed at the OLT each time a report is received from an ONU to enable centralized-based offloading and task distribution. The first

segment of the algorithm (lines 2-11) is the normal bandwidth allocation segment, except that each ONU will now be reporting the offloaded data size $R_{i,off}$, that is queued in its buffer, along with the amount of its regular upstream traffic R_i. The OLT grants each ONU bandwidth accordingly and schedules its next upstream transmission at $T_{i,start}$. The second segment of the algorithm (lines 12-30) focuses on edge traffic exchange. The OLT must extract information regarding the ONU's computational availability, whether it is currently transmitting edge data, and whether this data is an offloaded task or a computational result. Once edge computations are completed, results are forwarded to the origin ONU, whereas new tasks are assigned to ONUs with available resources.

4.4 Decentralized-Based Edge Computing

In this chapter, we use the same decentralized scheme used in Chapter 3, namely *Taking Turns with Adaptive Cycle Time* (TTACT), as the candidate for decentralized bandwidth allocation. We first take a closer look at the *out-of-band* (OOB) signaling scheme, which is used for managing upstream media access, before proposing a novel OOB signaling scheme that allows ONUs to have potentially larger windows for offloading.

4.4.1 TTACT with Immediate Tagging (IT)

In previous chapters, it was shown how the additional channel in TTACT is used to manage upstream media access among the ONUs. The channel allows each ONU to send a very short time-stamped frame (a tag) at the beginning of its transmission, announcing how many bytes it intends to send on the upstream. Chances of upstream idle gaps are significantly reduced since the time it takes the tag to reach the following ONU on the control channel will be during ongoing data transmission on the upstream channel. However, gaps may still exist in cases when there is no or very short upstream data to send.

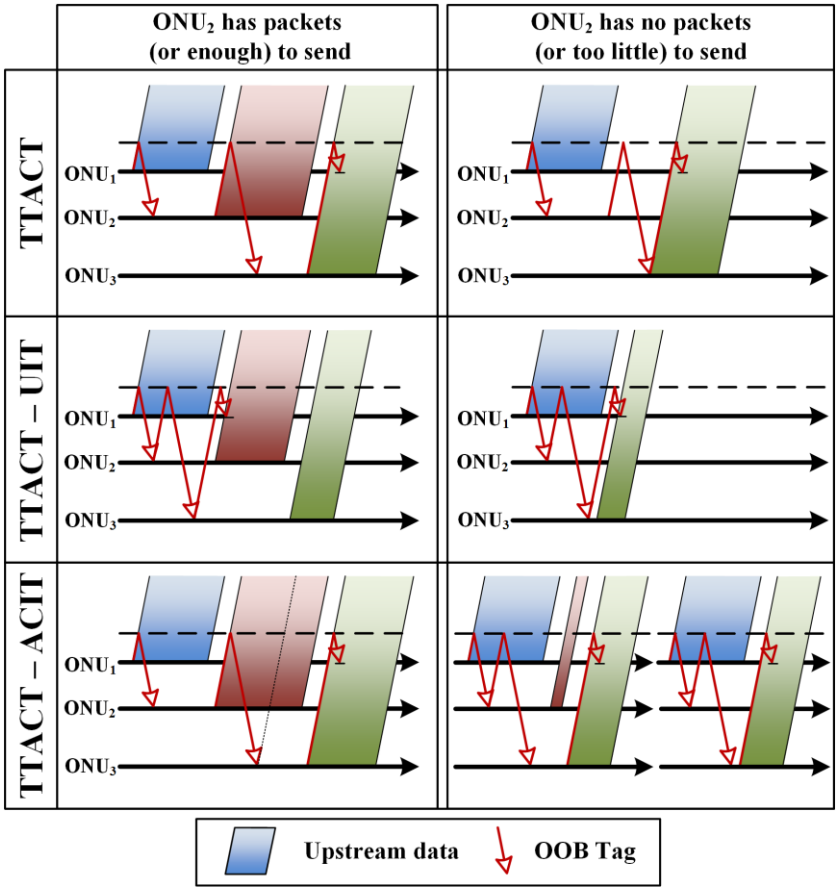


Figure 4.4: Normal tagging vs. immediate tagging.

In [29], several OOB tagging schemes were studied and evaluated to further improve the bandwidth allocation performance of TTACT. The schemes were based on decoupling the time of sending a tag from the time of upstream transmission. For instance, if a given ONU (ONU_i) has no upstream traffic to send, it does not have to wait until the start of its transmission window to tag the following ONU (ONU_{i+1}) and give up its turn. As illustrated in Figure 4.4, immediately sending the tag would allow the following ONU to access the medium right away and reduce any idle time gaps on the upstream channel. This principle has been referred to as immediate tagging, where the two most promising tagging schemes were found to be *unconditional immediate tagging* (UIT) and *aware conditional immediate tagging* (ACIT) [29].

In UIT, once an ONU receives a tag, it immediately tags the following ONU indicating how much it intends to send. Only reported packets may then be sent even if more packets had arrived before its start time of transmission. While this significantly compresses the cycle duration and reduces packet delays under light loads, it does not perform well under heavy loads (above 75%). ACIT, on the other hand, performs better under heavy loads while performing near optimum under light loads. This is because, in ACIT, an ONU may not tag the following ONU immediately upon being tagged if it has enough packets to utilize the propagation distance to the following ONU. Sending the tag later enables it to accommodate more newly arriving packets before its start time of transmission.

In the following sections, we propose novel tagging schemes that aim to better support exchanging edge traffic on the additional channel.

4.4.2 Edge Computing with TTACT

Inter-ONU communications in such a decentralized scheme enable the service composition and distribution of subtasks to be carried out in a different manner. Once ONU_c becomes the composition manager, three phases of communications along with a computational phase are proposed as follows:

Service Discovery and Selection Phase:

The main goal in this phase is to select the S ONUs that will carry out the S subtasks. This can be achieved in one of two ways, either by selecting them from a volunteer pool V or from an availability pool A. In the former, ONU_c must first broadcast its service request to all the ONUs and wait for responses from ONUs volunteering to carry out the service. In order for an ONU to determine if it will volunteer, ONU_c must append the following information to its outgoing tag:

1. amount of each subtask in terms of computation cycles and storage space,
2. size of each subtask's input data (D),
3. its QoS constraints.

After receiving this tag, each ONU will check its availability of resources and respond in its next outgoing tag message if it has the requested services available. Finally, from the received $N - 1$ tags, ONU_c will be able to select the S ONUs out of the volunteering pool V.

In the latter scheme, ONU_c alternatively selects the S ONUs from an availability pool A without the need to first broadcast its requirements. This can be done by requiring all ONUs to continuously append their computational status and offered services in any outgoing tag message similar to what is proposed to be done in REPORT messages in the centralized configuration. Using this information, ONU_c directly selects S ONUs by checking its received tags and choosing from the services offered. This would require each ONU to continuously store the availability information received in the last $N - 1$ tags, but will save the time wasted in requesting resources and waiting for volunteers. Possible selection criteria of the S ONUs can then be based on:

1. the first ONUs to respond in pool V (in the former scheme) or ONUs with the most recent tags in pool A (in the latter scheme),
2. ONUs that offer the service locally rather than on connected devices (less latency and higher security),
3. ONUs with least upstream traffic (since results can be retrieved faster using their excess bandwidth in case of in-band inter-ONU communications),
4. devices that have acceptable battery levels.

After the S ONUs are selected, ONU_c will then broadcast this selection in its next tag along with the subtask of each selected ONU.

Task Offloading/Distribution Phase:

In this phase, ONU_c broadcasts the offloaded data to the selected ONUs. Unlike the centralized scenario, data here can be directly sent to ONUs on the additional channel, instead of first having to send it to the OLT.

Service Execution Phase:

In this phase, each selected ONU processes its received task data and carries out the assigned subtask. During this period, selected ONUs may refuse to accept new services/tasks from connected devices (or from other ONUs) if they have no available resources. This can be reflected by no longer volunteering or by indicating a busy status in their outgoing tag messages.

Results Retrieval Phase:

After finishing its computation, each ONU sends its output back to ONU_c, which then combines all the results together and sends the full task result to its connected client.

Algorithm 2 presents the proposed algorithm that is to be carried out at each ONU to enable decentralized media access and exchanging edge data. The algorithm is provoked each time an ONU receives a new tag. If the tag belongs to the previous ONU in the transmission sequence, the ONU schedules its next upstream transmission at $T_{i,start}$ using the information extracted from the tag. Each ONU, however, does not calculate its next transmission duration W_i until it sends its tag (at $T_{i,TAG}$). This is because, by that time, new packets may have arrived and should be accounted for in the announced window as long as they still fit in the ONU's maximum allowable window $W_{i,\max}$. The upstream media-access portion of the algorithm is thus twofold; scheduling the transmission time (lines 5-9) and calculating its duration before sending the tag (lines 35-40).

Algorithm 2 Decentralized algorithm executed by ONU_i

```
1   if (TAG_j is received) then
        Extract and store computational info:
2       C_j, L_j ← extract computational availability and offered services
3       F_j ← extract if edge traffic exchange is imminent
4       A_j ← extract if upcoming edge data is request or output
        Schedule next upstream transmission if TAG_{i-1} is received:
5       if (j = i – 1) then
6           T_{i–1,start}, W_{i–1}, CLK_{i–1} ← extract ONU_{i-1} transmission info
7           calculate T_{i,start} and schedule upstream transmission
8           Generate TAG_i frame
9           TAG_i ← append {T_{i,start}}
            Determine tag scheme and computational parameters:
10          if (no edge data is queued in buffer) then
11              F_i = 0 ← reset offloading flag
                Check N – 1 offloading flags:
12              if (ΣF_{k≠i} > 0) then
13                  UITR/UITX ← switch to offloading tagging
14              else if (ΣF_{k≠i} = 0) then
15                  ACIT/UTIR ← switch to normal tagging
16              end if
17          else if (edge data is queued in buffer) then
18              if (data has not been previously announced) then
19                  F_i = 1 ← set offloading flag to imminent
20                  UITR/UITX ← switch to offloading tagging
21                  schedule transmission at next OOB window
22                  if (data is offloaded from connected client) then
23                      S ← check if task is dividable
24                      A ← check N – 1 availability flags
25                      S ← select S ONUs from availability set A
26                  TAG_i ← append {S, task_ID, subtask_IDs, size(D)}
27                  end if
28              else if (data has been previously announced) then
29                  compute OOB slot duration: (C_avg – T_tags)/ΣF_k
30                  send edge data at OOB slot
31              end if
32          end if
33          C_i, L_i ← compute computational availability flags
34          TAG_i ← append {C_i, L_i, F_i}
            Send TAG_i:
35          T_{i,TAG} ← compute outgoing tag time in current scheme
36          if (T_{i,TAG} = CLK_i) then
37              W_i = max(BUF_i, W_{i,max})
38              TAG_i ← append {W_i, CLK_i}
39              send TAG_i
40          end if
41      end if
42  end if
43  if (edge data is received with ONU_i as destination) then
44      if (task/subtask has been assigned) then
45          start execution once data is fully received
46          buffer results once computation is complete
47      else if (results from previously assigned ONU) then
48          forward to client once data is fully received
49      end if
50  end if
```

The rest of the algorithm deals with the exchange of edge data, where any offloading ONU is responsible for selecting the S ONUs based on the availability status extracted from the last $N-1$ tags. Moreover, depending on whether or not edge data is to be exchanged, the tagging scheme is switched, as will be discussed in the following section.

4.4.3 Proposed OOB Tagging for Edge Traffic

The tagging scheme can be modified to better support exchanging edge data on the additional channel without affecting the signaling required for managing upstream traffic. In this section, we introduce two novel tagging schemes that allow for larger edge traffic windows before discussing the proposed tag frame structure in the following section.

UIT with Reset (UITR):

The difference between this tagging scheme and the original UIT, introduced in [29], is that UIT here is reset at the beginning of each cycle. In other words, the first ONU in the sequence here must wait until the start of the new cycle before sending out its tag, whereas each of the other ONUs sends out its tag once it receives one from its preceding ONU. This does not shrink the cycle as much and still performs at optimal levels under both heavy and light loads. More importantly, it allows a silent period to exist on the OOB channel from the time tags are exchanged (at the beginning of each cycle) until the end of the cycle. This silent period can then be used for exchanging edge traffic.

UIT with cycle eXtention (UITX):

In this tagging scheme, the cycle is allowed to extend beyond the ONUs' self-allocated upstream transmissions windows. This is very useful, under light network loads, when the cycle

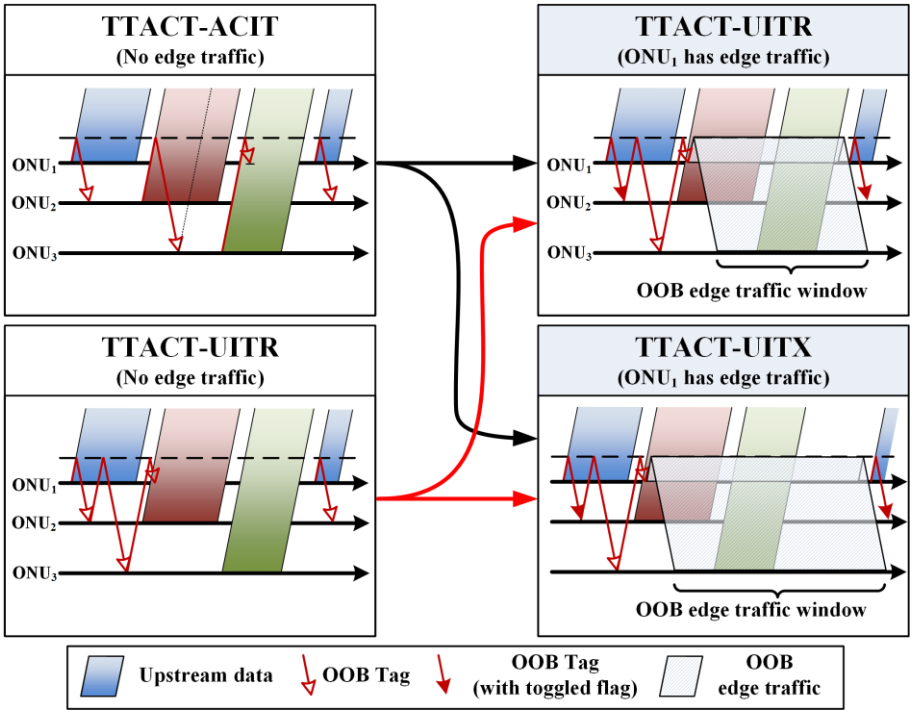


Figure 4.5: Proposed tagging schemes for normal and offloading states.

may not be long enough to allow for a viable OOB silent period for exchanging edge traffic. Therefore, during edge traffic exchange, the cycle is allowed to extend as follows:

$$C = \begin{cases} C, & \text{if } C > XT_{tags} \\ \min\left(XT_{tags}, C_{\max}\right), & \text{otherwise} \end{cases} \tag{4.6}$$

where X is the cycle extension factor ($X > 1$).

The common characteristic between UITR and UITX is that their tagging is done at the beginning of each cycle, leaving a subsequent OOB silent period that can be used for exchanging edge traffic. Such a period does not exist in any of the immediate tagging schemes introduced in [29]. As illustrated in Figure 4.5, we propose a hybrid tagging arrangement for exchanging edge traffic using these new tagging schemes as follows:

1. the default tagging scheme will be set as ACIT or UITR since they have similar near-optimal performances,

2. once any given ONU has edge traffic to send, it toggles a certain flag in its next tag,
3. this flag switches the default tagging scheme to either UITR or UITX, forcing all ONUs to immediately send out their tags, once they receive one, at the beginning of each cycle,
4. since there may be more than one ONU ready to transmit OOB edge traffic, the free window in the OOB cycle is then divided among ONUs with toggled flags (line 29 in Algorithm 2),
5. as long as this flag is toggled, in at least one of the previous N tags, the tagging remains UITR/UITX (lines 12-13 in Algorithm 2), otherwise, the tagging scheme switches back to the default scheme whether it was ACIT or UITR (lines 14-16).

That way, the overall performance is expected to be near-optimal by using ACIT/UITR most of the time and only switching to UITR/UITX when edge traffic needs to be transmitted.

4.4.4 Tag Frame Structure

Figure 4.6 illustrates a proposed frame format for the tags exchanged by the ONUs in the decentralized scheme. It is based on the frame structure of the *multipoint control protocol* (MPCP) of the EPON standard (IEEE 802.3ah) but is modified to support both edge traffic exchange and service composition in addition to managing upstream media access. The proposed frame contains the following fields;

- the destination and source ONU addresses,
- a frame type ID (indicating that it is a decentralized control frame),
- a code indicating the type of control frame (i.e. tag),
- a local timestamp,
- the ONU's next transmission window size (W_i) and start time ($T_{i,start}$),
- 4 flags and a reserved task ID field (if exists),

Field					Octets (Bytes)
Destination Address (DA)					6
Source Address (SA)					6
Frame Type = control frame					2
Opcode = ONU tag					2
Timestamp					4
W_i					2
$T_{i,start}$					2
C	*L*	*F*	*A*	Task ID	1
Size(D/R)					2
Selected ONU ID #1					1
Subtask ID #1				Size(E_1)	1
Selected ONU ID #2					1
Subtask ID #2				Size(E_2)	1
⋮					⋮
Selected ONU ID #16					1
Subtask ID #16				Size(E_{16})	1
Pad = 0					1
Frame Check Sequence (FCS)					4

Figure 4.6: Proposed tag frame structure for supporting edge computing in TTACT.

- the size of the subtask input edge data (if exists),
- fields reserved for assigning subtasks (up to 16) or indicating offered services,
- and a frame check.

The 4 flags used are;

- *C*: indicates that computing resources are available if set to 1,
- *L*: indicates that computing resources are local if set to 1,
- *F*: if set to 1, indicates that edge traffic is imminent and forces ONUs to switch the offloading tagging scheme,
- *A*: indicates that selected ONU IDs and their subtasks are included in this tag if set to 1.

4.5 Edge Computing Delay Analysis

The edge delay performance is different in each allocation paradigm. In this section, we examine the delay components of each one through an analytical framework similar to the one introduced in the previous chapter.

4.5.1 In Centralized Multi-thread Polling

Because ONUs have no direct link between them in the centralized case, edge and service computing delays become highly dependent on the RTTs. Upon receiving a service request, ONU_c will first request service allocation from the OLT. The average time it would take ONU_c to send its request, in its next report, would approximately be:

$$T_1 \approx C_{avg} / 2M + C_{avg} \tag{4.7}$$

The expression above resembles the average reporting and grant delays, illustrated in Figure 4.3, which are approximated in terms of the average cycle duration C_{avg} and the number of threads M. After receiving the request, the OLT will start granting ONU_c more upstream bandwidth, up to $W_{c,\max}$ along with a portion of excess bandwidth borrowed from lightly-loaded ONUs (see [18] for details on multi-thread excess distribution). This offloading lasts for N_D cycles and takes some time that is on average:

$$T_2 \approx N_D(C_{avg} + M(W_{c,\max} - W_{c,avg}) + W_{c,excess}) + \frac{RTT_c}{2} \tag{4.8}$$

where $W_{c,avg}$ here reflects the average transmission window used by ONU_c for upstream traffic, $W_{c,\max} - W_{c,avg}$ represents the average unutilized window per thread that can be used to transmit offloaded data, and $W_{c,excess}$ represents the excess bandwidth portion per cycle allocated to ONU_c for offloading. While data is being offloaded, the cycle duration will extend by:

$$\Delta T_{off} = C_{off} - C_{avg} = M(W_{c,\max} - W_{c,avg}) + W_{c,excess} \tag{4.9}$$

This cycle extension represents the average allocated bandwidth for offloading. The number of cycles needed for offloading data D can then be expressed as:

$$N_D = \frac{D / R_{up}}{M(W_{c,\max} - R_c) + W_{c,excess}} = D / R_{up}\Delta T_{off} \tag{4.10}$$

where R_{up} is the upstream data rate. After receiving the data, the OLT chooses S ONUs, assigns them their subtasks in their next grants, and sends their input data. On average, this would take:

$$T_3 \approx T_{process}^{OLT} + D / R_{down} + C_{avg} / M + RTT / 2 \tag{4.11}$$

where the third term reflects the average thread duration, and the last term is the average propagation delay. The total task distribution delay is then the sum of T_1, T_2, and T_3, which, assuming $RTT \approx RTT_c$, can be expressed as:

$$T_{off} \approx (\frac{3}{2M} + N_D + 1)C_{avg} + RTT_c + \frac{D}{R_{up}} + \frac{D}{R_{down}} + T_{process}^{OLT} \tag{4.12}$$

After the subtasks are distributed, the OLT waits for each ONU to finish its computation and upload the output using its extra-allocated bandwidth. The cycle extension here may be longer than the extension experienced in the offloading phase if $S \geq 1$. This is because S ONUs will be using their extra bandwidths to send their computational results. Assuming that the ONUs finish their computations at the same time, retrieving their task results will extend the cycle by:

$$\Delta T_{ret} = C_{ret} - C_{avg} = \sum_{i \in S} M(W_{i,\max} - W_{i,avg}) + W_{i,excess} \tag{4.13}$$

Although the cycle is shown to extend more in the retrieval phase, its duration may not last as long as the offloading phase. This is because edge data here may be smaller for each ONU. In other words, the total computational output is transmitted using the excess bandwidth of S ONUs compared to only using the excess bandwidth of ONU_c as done in the offloading phase. The number

of cycles needed to upload the computational output may therefore be less but may still differ for each of the S ONUs depending on the size of its task output E_s and its available bandwidth. The number of cycles needed to upload ONU$_i$'s computational result, where $i \in S$, is then:

$$N_{i,E} = \frac{E_s / R_{up}}{M(W_{i,\max} - W_{i,avg}) + W_{i,excess}} \tag{4.14}$$

To arrive at an approximation for the average retrieval delay, we assume computational outputs and available bandwidth for each of the S ONUs to be equal in size. That way, the result-retrieval delay can be expressed as:

$$T_{ret} \approx (\frac{3}{2M} + N_E + 1)C_{avg} + RTT + \frac{SE}{R_{up}} + \frac{SE}{R_{down}} + T_{process}^{OLT} \tag{4.15}$$

where N_E is the number of cycles it takes to upload E bytes from any of the S ONUs.

4.5.2 In Modified TTACT

In the decentralized scenario, the time it would take ONU$_c$ to send its tag announcing the selected S ONUs is:

$$T_1 \approx C_{avg} / 2 \tag{4.16}$$

Note however that the time to send the tag is now independent of the ONU's turn to transmit on the upstream channel but rather depends on the selected tagging scheme. After announcing ONU$_c$'s selection, ONU$_c$ will start offloading in the next unused silent OOB period, which is proportional to the cycle length. This would take N_D cycles and can be expressed as:

$$N_D = \frac{D / R_{OOB}}{C_{avg} - T_{tags}} \tag{4.17}$$

where T_{tags} is the duration in which tags are exchanged between ONUs on the control channel at the beginning of each cycle, while (C_{avg} – T_{tags}) represents the silent period that will be used to

transmit edge data D. Before ONU_c can start transmitting offloaded data on the OOB channel, it must first wait until tags are exchanged between the ONUs. This delay, including offloaded data transmission, would then become:

$$T_2 \approx T_{tags} + N_D C_{avg} \tag{4.18}$$

The total offloading delay is, therefore:

$$T_{off} \approx \frac{C_{avg}}{2} + T_{tags} + N_D C_{avg} + RTT_{SC} \tag{4.19}$$

where RTT_{SC} is the average round trip time that the OOB signal takes to reflect back and reach other ONUs. Once computations are performed, each selected ONU would start sending its result, which would take:

$$T_{ret} \approx \frac{C_{avg}}{2} + T_{tags} + N_E C_{avg} + RTT_{SC} \tag{4.20}$$

where N_E can be expressed as:

$$N_E = \frac{SE / R_{OOB}}{C_{avg} - T_{tags} - ST_g} \tag{4.21}$$

Note that the OOB edge transmission window would now have to be divided among the S ONUs assuming they finish their tasks around the same time. Each ONU can simply calculate its OOB window length by dividing the free window ($C_{avg} - T_{tags}$) among the ONUs possessing edge traffic (line 29 in Algorithm 2) before scheduling its OOB transmission based on its place in the transmission sequence.

4.6 Numerical Results

In this section, we first compare the offloading delays of centralized and decentralized bandwidth allocation before examining the factors that affect their edge computing performance.

We consider an asymmetric 10G-EPON with a 100km reach consisting of an OLT and 16 ONUs. The ONUs share an upstream wavelength of 1Gbps, whereas from the access side the end-users have an access rate of 100Mbps. The RN is assumed to be located 95km away from the OLT, where the FBG for the decentralized scheme is also located. ONUs are again placed randomly in the last 5km of a 100km network span with each ONU having a 10MB buffer. The inter-transmission guard interval T_g is set to 5µs. For multi-thread polling, we use 3 threads with the same parameters used in [18]. For the proposed decentralized scheme, we consider the OOB loop-back technique with a rate of 1Gbps. We use UITR tagging in the default state combined with UITR/UITX tagging in the edge transmission state.

4.6.1 General Offloading Performance

Figures 4.7 and 4.8 illustrate the offloading and retrieval delays of a 5MB task that is divided among 5 ONUs with a total output data of 1.5D. Edge traffic delays are shown to increase with upstream traffic load in centralized multi-thread polling, especially at loads greater than 80%. This is because, as the network load increases, the unused bandwidth in the ONU's window decreases. Exchanging edge traffic would then take longer and last for more cycles. On the other hand, edge delays within the decentralized schemes decrease as the load increases. This is because the more the cycle is extended, the larger the OOB window becomes. UITR is shown to perform much worse than UITX under light loads since UITR's cycle is too small and barely extends beyond the duration necessary for tagging ($C_{avg} \approx T_{tags}$). Extending the cycle in UITX therefore greatly improves the performance under light loads. The figures also compare simulation delays against those calculated from the analytical expressions derived in Section 4.5, namely Eqs. (4.12), (4.15), (4.19), and (4.20). The average cycle durations obtained from the simulations are used to calculate delays analytically, which are then found to accurately capture edge delays in all schemes.

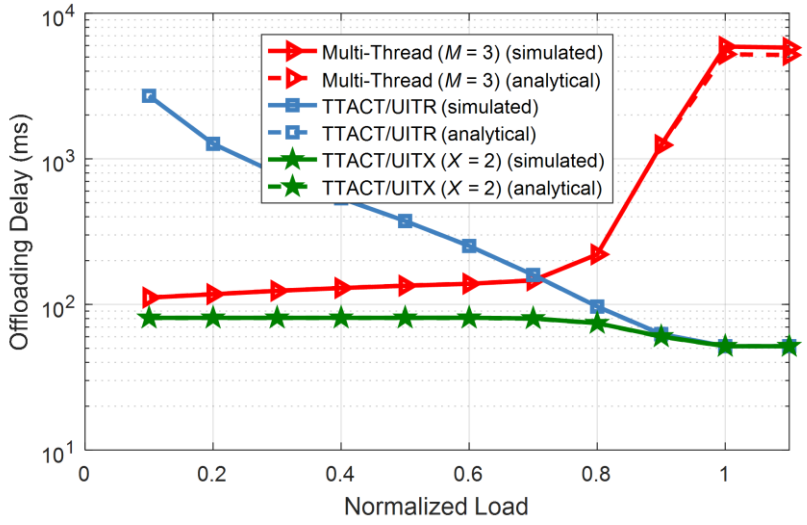


Figure 4.7: Analytical and simulated task-offloading delays.

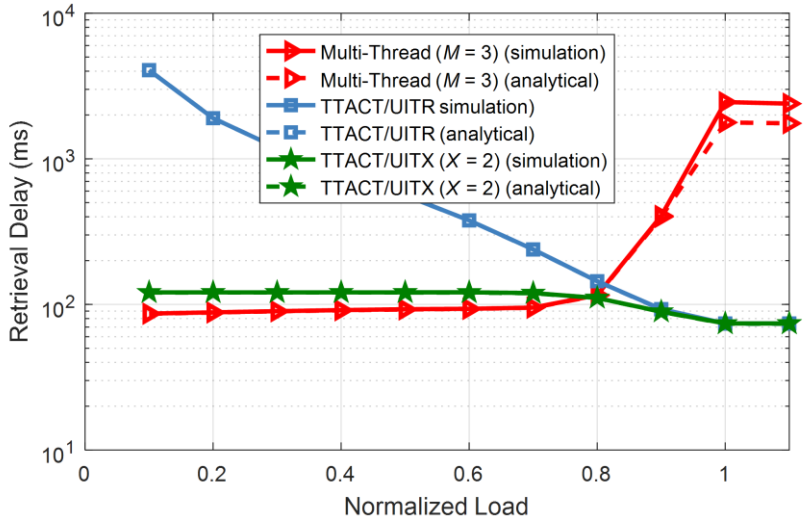


Figure 4.8: Analytical and simulated result-retrieval delays.

Figures 4.9 and 4.10 show the relative error percentages between simulated and analytically computed delays, which are found to be below 1% under normal loads.

Looking again at Figures 4.7 and 4.8, notice how retrieval delays in multi-thread polling are relatively lower than its offloading delays, whereas it is quite the opposite in decentralized schemes. This is because, in the centralized retrieval phase, there is no bandwidth contention

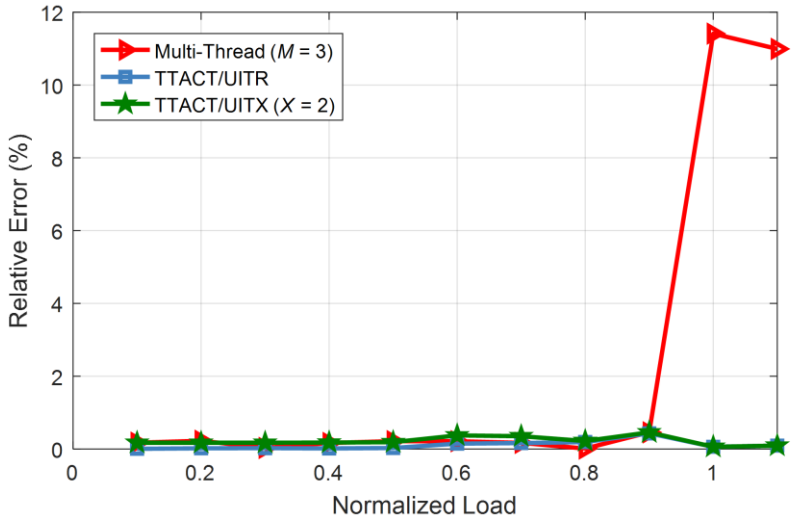


Figure 4.9: Relative error of task-offloading delays.

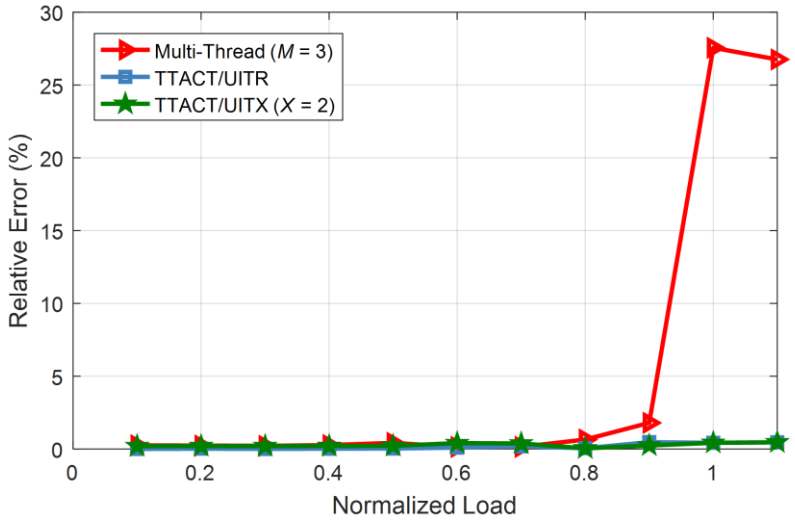


Figure 4.10: Relative error of result-retrieval delays.

between ONUs. Each of the S ONUs uses its own excess to send its result. Sending the output data back to ONU_c is thus done using the resources of the S ONUs compared to only using the available bandwidth for ONU_c during the offloading phase. However, in the decentralized case, the S ONUs contend for the silent OOB window in the retrieval phase and must then leave guard intervals between their OOB transmissions. This does not occur in the offloading phase, during which ONU_c

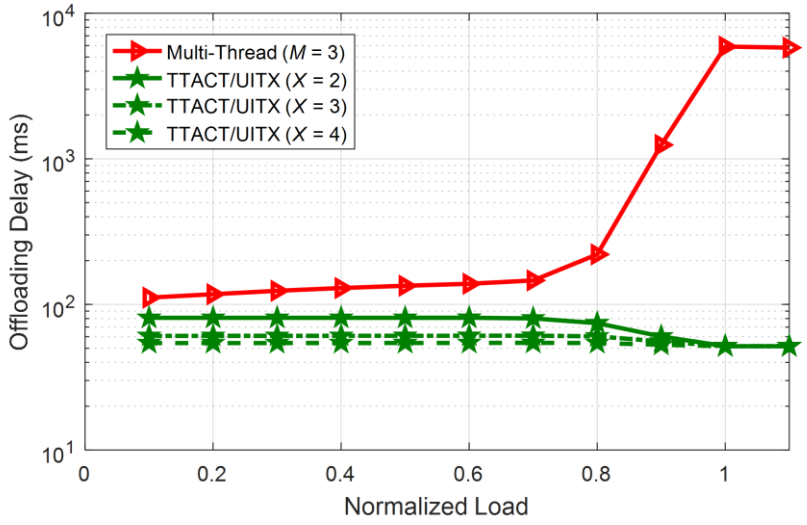


Figure 4.11: Effect of extending the decentralized cycle duration.

uses the full silent window by itself. Delays in the decentralized scheme are therefore affected by concurrent usage of the OOB window.

4.6.2 Effect of Extending the Decentralized Cycle

Figure 4.11 shows the effect of allowing the decentralized cycle duration to extend more than double its minimum as expressed by Eq. (4.6). Preemptively extending the cycle during edge traffic exchange allows more traffic to be sent much sooner, especially under light loads, where the cycle is too small to allow for a viable OOB window. However, this extension has a negative effect on upstream traffic delays as further discussed below.

4.6.3 Effect of Edge Traffic on Upstream Traffic

Figures 4.12 shows the pre-transmission delays of regular upstream traffic during both an offloading phase and a retrieval phase. Injecting edge traffic on the upstream wavelength has a significant effect on multi-thread polling, especially during a retrieval phase where multiple ONUs

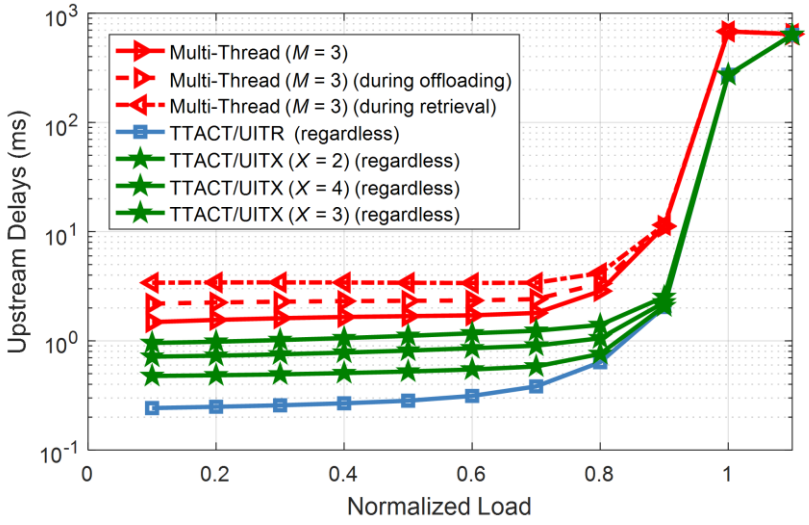


Figure 4.12: Effect of edge traffic on upstream traffic delays.

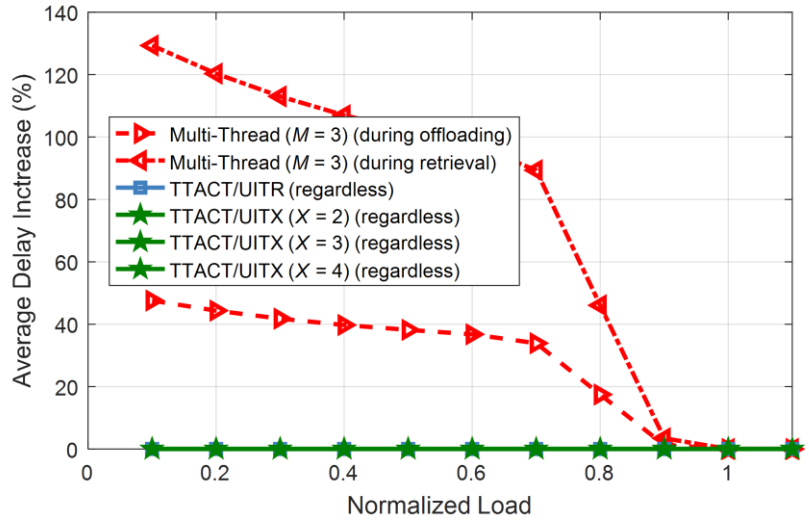


Figure 4.13: Delay increase of upstream traffic delays.

may be transmitting. This is further illustrated in Figure 4.13, which shows the percentage increase in the upstream delays. The delays of centralized multi-thread polling increase by more than double under light loads in the retrieval phase since injecting edge traffic considerably extends its cycle and causes further delays for upstream packets. Even though the delays of the decentralized schemes are not affected by exchanging edge traffic, switching from UITR to UITX will have

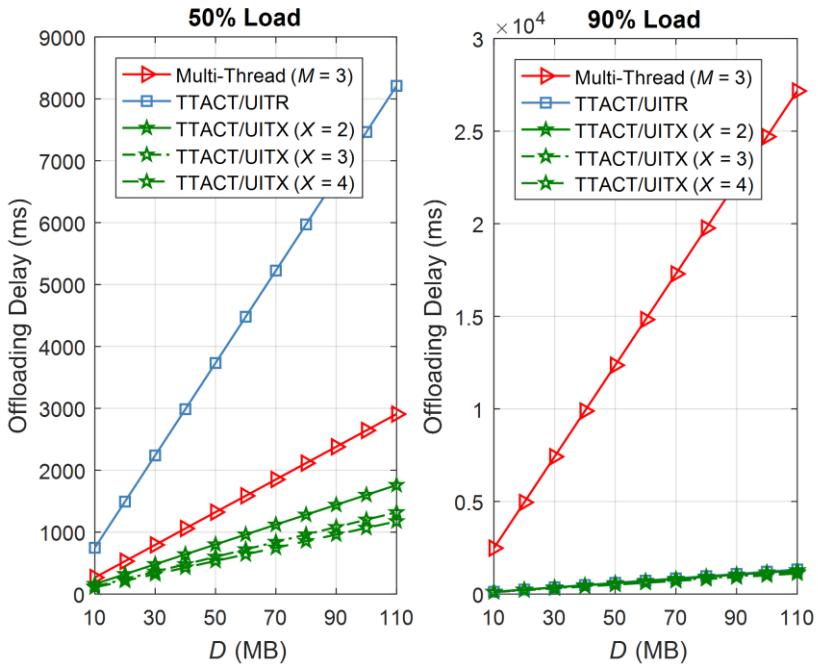


Figure 4.14: The effect of offloaded data size on offloading delays.

negative side effects on upstream packet delays due to extending the decentralized cycle duration. These negative effects on upstream delays, however, will only last while there is ongoing edge traffic transmission for all schemes. The overall performance would then depend on how often the network has to deal with edge traffic and the size of that traffic.

4.6.4 Effect of Offloaded Data Size on Offloading Delays

Figure 4.14 illustrates the effect of the size of the offloaded data on offloading delays. Under moderate loads, both UITX and multi-thread polling are able to perform well, whereas UITR does not have enough OOB bandwidth availability to transfer edge traffic right away with its short cycle duration. Under heavy loads, however, both decentralized schemes perform better than multi-thread polling since their cycle extends towards its maximum, enabling more OOB edge traffic to be sent. The performance of centralized multi-thread polling, on the other hand, tends to

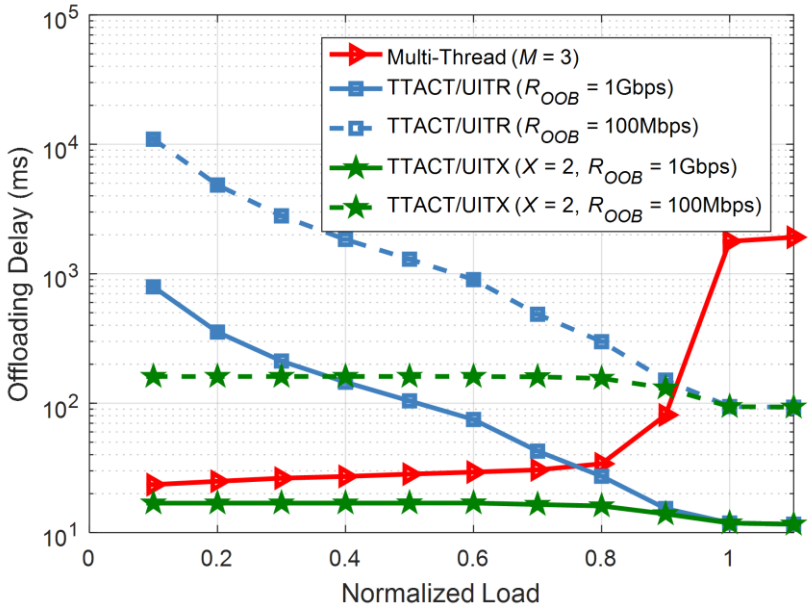


Figure 4.15: The effect of the OOB rate on decentralized offloading delays.

significantly worsen for the lack of excess bandwidth in the ONUs' granted windows, especially with large offloaded data.

4.6.5 Effect of the OOB Channel Rate

Figure 4.15 shows how changing the rate of the OOB control channel effects decentralized edge delays. In order to take full advantage of the decentralized scheme, the OOB rate must be at least comparable to the upstream rate (i.e., 1Gbps).

4.7 Conclusions

In this chapter, we examined the feasibility of cooperative edge computing in LR-PONs. We investigated the delays experienced in exchanging edge traffic as well as the side effects on upstream traffic. We compared between centralized multi-thread polling against a proposed modification to the decentralized scheme. We believe that only the latter can support true edge

Table 4.2 Comparison of Centralized and Decentralized-Based Service Composition

	Centralized	Decentralized
Composition manager	OLT	client ONU or elected ONU
Composition delay	long	short
Delay mostly affected by	network span	concurrent edge transmissions (due to sharing OOB channel)
Effects on regular traffic	high (exaggerated with concurrent edge transmissions)	negligible
Offered services	network wide	local ONUs only
Service rejection ratio	potentially low	potentially high

computing in LR-PONs since it requires no OLT involvement and achieves lower delays for both edge and non-edge traffic. Decentralized-based service composition also has no side effects on regular upstream traffic except when deliberately extending the cycle to allow for longer OOB windows for edge traffic. This, however, comes at the cost of placing additional transceivers within the ONUs and modifying the architecture to allow inter-ONU communications to take place. Moreover, the ONUs themselves have to select the composition manager and run relatively more complex algorithms than ONUs in a centralized-based scheme. On the other hand, centralized schemes may still offer some benefits for service composition despite their long delays. For instance, the OLT can easily gain access to ONUs in other access zones, to which it may choose to forward service requests instead. Centralized-based service composition may also offer lower service rejection ratios as well as additional services that are only available in other access zones. Table 4.2 highlights the differences between centralized and decentralized-based service composition and the strengths of each.

Chapter 5

Dynamic and Energy-Efficient Bandwidth Allocation in EPONs

In this chapter, we shift our focus to the energy-efficiency requirement in optical access networks, which is typically acquired by employing a cyclic sleep mode for the ONUs. We study the tradeoffs in the network performance when employing energy conservation schemes and develop novel schemes that aim to enhance the performance by making the bandwidth allocation more dynamic. In the first part of the chapter, we focus on centralized upstream allocation, for which we propose and examine different schemes that aim to improve its network performance while keeping its energy consumption as low as possible. In the second part of the chapter, we propose a novel decentralized framework that aims to conserve energy and have comparable network performance with decentralized TTACT.

5.1 Proposed Centralized Energy-Conserving Framework

In Chapter 2, we introduced fixed-slot allocation, which is chosen by many studies for energy conservation in PONs. In fixed-slot allocation, the ONU receives its downstream traffic during its upstream transmissions in a designated slot. In other words, downstream transmissions are regulated and scheduled to overlap upstream transmissions. This allows each ONU to switch

off its transceiver outside its slot. Moreover, if the ONU's upstream transmission finishes before that of the OLT within the slot, it may enter into doze mode only turning off its transmitter. Once the slot ends (or if both transmissions finish early), the ONU may immediately switch to sleep-mode setting a timer for its next wake up for its scheduled slot in the next cycle. While in sleep-mode, the ONU still buffers any received upstream packets from its connected devices. Similarly, the OLT buffers any received downstream traffic addressed to the sleeping ONU until its next slot. Even though fixed-slot allocation may reduce the ONUs' energy consumption, it suffers from two main drawbacks; poor utilization and no allocation differentiation between heavily and lightly-loaded ONUs. In other words, the bandwidth allocation is reduced from its dynamic nature to a static one, causing heavily-loaded ONUs to experience longer delays and potential frame losses. In this section, we propose new schemes that aim to make fixed-slot allocation more dynamic.

For the inner workings of fixed-slot allocation, we assume that the OLT specifies an ONU's sleep duration (or its next wake-up time) in the same GATE message the ONUs receives at the beginning of each slot, informing it of the start time and duration of its upstream transmission. The sleep duration then depends on the cycle duration and the wake-up overhead, which can be expressed as:

$$T_s \geq (1-1/N)C_{\max} - T_{oh} \qquad (5.1)$$

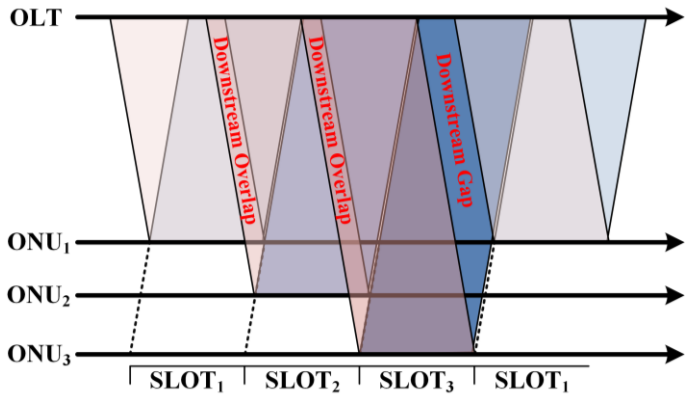
where $C_{\max}$ is the maximum cycle duration, N is the number of ONUs, and T_{oh} is the wake-up overhead (~2ms for a 1Gbps transceiver). Note that a sleep duration can only exist if the ONU's idle time is greater than the wake-up overhead. Note also that the sleep duration may be greater than the duration on the right-hand side of the expression by the amount of the additional idle time that may exist in the ONU's slot itself if the traffic exchange finishes early.

5.1.1 Proper Downstream-Upstream Locking

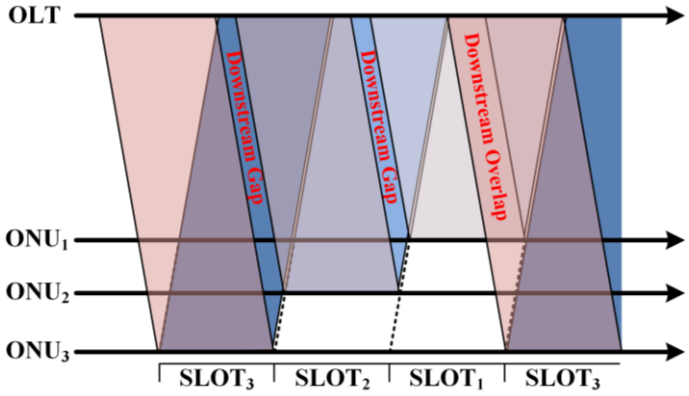
The key to achieving energy savings in PONs is through minimizing the ONUs' awake times and maximizing their sleep times. This led to combining downstream and upstream transmissions for each ONU in a single dedicated slot to confine its awake time to only its designated slot. This downstream-upstream locking, however, presents two problems for downstream transmissions.

The first problem is the limiting of downstream transmissions, which are usually much greater in size than upstream transmissions. Even when other ONUs are not fully using their downstream bandwidth, this type of allocation may prevent the ONU from receiving additional downstream frames that exceed its slot duration. One possible solution to this problem is to make use of asymmetrical PON rates, where the downstream transmission rate can be ten times the upstream rate. A similar slot duration would then mean different transmission sizes for each. Such a solution would not require ONUs to stay awake more than their designated slots.

The second problem that arises from locking downstream to upstream transmissions within the same slot is the resulting inter-transmission gaps and overlaps. Figure 5.1 illustrates these side effects for both cases; when using a near-to-far or a far-to-near ONU sequence. As illustrated in the figure, there would be either an overlap or a gap duration between ONU_i and ONU_{i+1}'s downstream transmissions for each case. This duration would be $2(L_{i+1} - L_i)/c$, where L_i is the distance of ONU_i from the OLT and c is the speed of light in the fiber. At the end of the cycle, there would also be a gap or overlap duration equal to $2(L_N - L_1)/c$. To the best of our knowledge, this problem has not been addressed in any related study even though downstream-upstream locking has been extensively used. A simple solution for the first case would be to decrease the downstream duration of each ONU by the overlap duration at the end of its slot. A possible solution



a) closest-to-furthest ONU transmission sequence



b) furthest-to-closest ONU transmission sequence

Figure 5.1: Downstream-upstream-locking issues: downstream overlaps and gaps.

for the second case would be to start the next cycle after the overlap duration at the end of the previous cycle. The resulting silent upstream period can then be used for other purposes such as new ONU discovery. In the rest of this work, we consider the closest-to-furthest assignment with downstream downsizing to eliminate transmission overlaps.

5.1.2 Proposed Upstream Allocation Schemes

As mentioned earlier, fixed-slot allocation suffers from two main drawbacks; poor utilization and no allocation differentiation between heavily and lightly-loaded ONUs. In this section, we propose new schemes that aim to make fixed-slot bandwidth allocation more dynamic.

Fixed-Slot with Full-Credit (FS-FC):

In this allocation scheme, the OLT grants each ONU its whole slot duration regardless of its request. This is different from using fixed-slot with limited service, granting the ONU no more than its reported buffer status in its last REPORT. Full-credit would allow ONUs to accommodate newly arriving packets that had not yet been reported. This is expected to be very useful when long upstream cycle durations are used, which are preferable for energy conservation. In such long cycles, the reported ONU information would most likely be outdated and many new packets may have arrived at the ONU by the time the OLT sends its grant. This scheme, however, does not offer any advantaged for heavily-loaded ONUs already fully utilizing their granted slots.

Fixed-Slot with Gap-Filling (FS-GF):

With gap-filling, the OLT does not grant a lightly-loaded ONU more than its request (i.e. it uses limited service). It then takes note of the unutilized portion of its slot (the upstream gap) and grants it to a heavily-loaded ONU as an extra transmission window. Since a heavily-loaded ONU will normally be in sleep mode outside its fixed time slot, this window information must be sent within the same GATE message that the ONU receives at the beginning of its slot. The gap must therefore be available after the ONU's wake-up (i.e., in the following cycle) and its information must have already been known to the OLT before it sends the ONU's grant. The heavily-loaded ONU may then either stay awake, if the gap is available shortly after its time slot, or it may switch to sleep mode and wake up earlier in the next cycle if the gap belongs to one of its preceding ONUs in the transmission sequence. It is therefore expected that heavily-loaded ONUs, in this scheme, will use some additional power due to staying awake more often and potentially spending some idle time waiting for their extra allocated gaps.

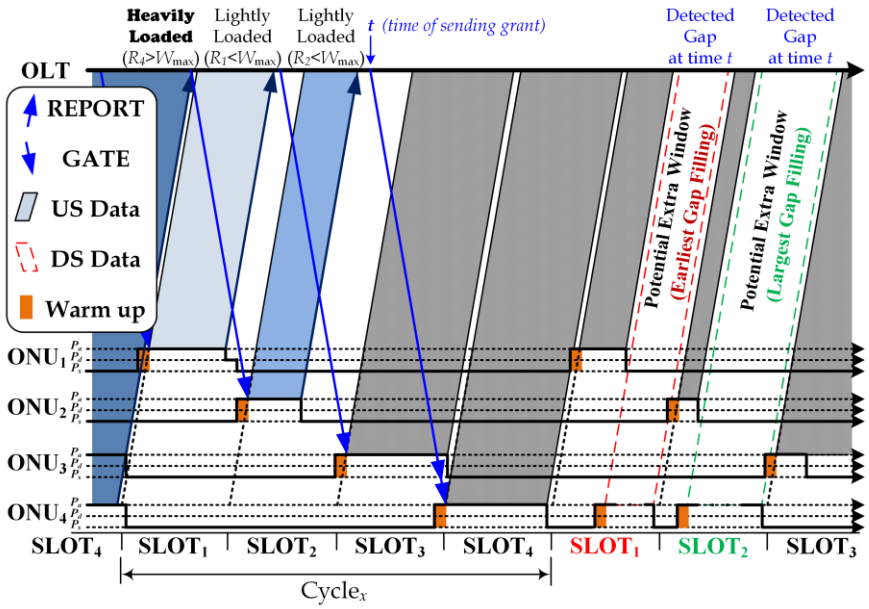


Figure 5.2: Fixed-slot with gap-filling

Figure 5.2 illustrates the gap-filling concept along with each ONU's transceiver power level. While unutilized bandwidth from ONU_1 and ONU_2 could not have been used in time to aid heavily-loaded ONUs in cycle *x*, their reports inform the OLT of their gaps in the next cycle and do so before time *t*, at which the OLT sends ONU_4's grant. With the OLT having a clear view of gaps within the next cycle, it sends ONU_4 a double grant: one for its normal slot and another for using either ONU1's or ONU2's gap depending on the gap-selection criteria, whether it is earliest possible or largest available as will be further discussed.

Fixed-Slot with Gap-Filling or Full-Credit (FS-GF/FC):

In this scheme, either gap-filling or credit allocation is used for each given slot as follows. Similar to the previous scheme, the OLT will make a record of any existing idle upstream gap belonging to a lightly-loaded ONU_i. If no heavily-loaded ONU has been allocated this gap, up until the time of sending ONU_i its grant, the OLT gives the gap back to ONU_i as full credit.

Fixed-Slot with Gap-Filling and Credit (FS-GFC):

In this scheme, the OLT grants lightly-loaded ONUs their requested windows along with a partial credit proportional to their request. It then reserves their remaining upstream gaps for heavily-loaded ONUs, similar to the previous scheme, and later returns any unclaimed gap to its original ONU as full credit if no heavily-loaded ONU claims it in time. This scheme may therefore be viewed as a compromise between FC and GF/FC.

We further divide gap-filling techniques into two categories according to the gap-selection criteria used by the OLT:

1. ***Earliest Gap-Filling*** **(EGF):** The OLT here allocates heavily-loaded ONUs the earliest possible gaps, given that a gap size is greater than a preset threshold value W_{th}. This aims to serve two main purposes: minimizing queued packet delays and decreasing ONUs' awake times as much as possible by not making them wait too long for their additional allocated upstream windows.
2. ***Largest Gap-Filling*** **(LGF):** The OLT's main concern here is to maximize utilization. It therefore targets the largest available gaps in each cycle and allocates them to the heavily-loaded ONUs. This also happens to substantially relieve heavily-loaded ONUs from their heavy traffic since they are granted sizable windows that may be as long as a whole slot belonging to another ONU having no upstream traffic.

5.1.3 Improvement Index

Most energy-focused PON studies use energy consumption and packet delays as their main performance metrics [39], [74], [77]–[80], [88]. However, because in this work we compare the proposed schemes to basic fixed-slot allocation, we devise a metric that reflects the performance

Table 5.1 Symbol Definitions and Simulation Values

Symbol	Description	Value
$L_{\max}$	Maximum network span	100km
$L_{i,SC}$	ONU$_i$'s distance from splitter/combiner	0–5km
L_{SC}	Splitter distance from OLT (feeder span)	95km
R_{down}	Downstream transmission rate	10Gbps
R_{up}	Upstream transmission rate	1Gbps
R_{user}	Connected users' upstream rate	100Mbps
N	Number of ONUs	16
Q	ONU buffer size	10MB
L	Normalized network load	0.1, 0.5, 0.9
$C_{\max}$	Max. cycle duration	10 – 90ms
T_g	Guard interval	5µs
T_{oh}	Warm-up overhead	2ms
W_{Th}	Gap-filling threshold	1000bytes

in terms of both delays and energy savings. This parameter helps in identifying the overall best scheme despite any tradeoffs. We call the proposed metric the improvement index, which can be regarded as a benefit-to-cost ratio. The benefit here is the delay-improvement (i.e., the reduction in delays) while the cost is the additional power consumption (i.e., the additional awake time required). We express the improvement index I for any scheme j as:

$$I_j = \frac{(D_{FS} - D_j) / D_{FS}}{E_j / E_{FS}} \tag{5.2}$$

where D_{FS} and E_{FS} are the average delays and power usage of conventional fixed-slot allocation, respectively, whereas D_j and E_j are the average delays and power usage when using scheme j. The reason for using the power ratio in the denominator rather than the additional power percentage ($1 - P_j/P_{FS}$) is to be able to obtain an improvement index value for fixed-slot allocation. That way, the index for fixed-slot allocation becomes 0 and the best scheme will accordingly be the one having the largest index.

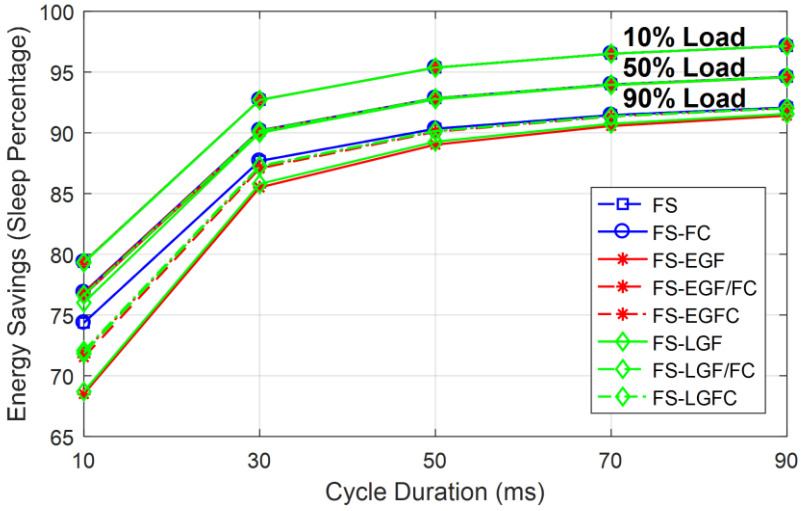


Figure 5.3: Energy savings vs. cycle duration.

5.1.4 Numerical Results for the Centralized Energy-Saving Framework

In our study, we consider a 100km long-reach EPON, in which the ONUs share an upstream wavelength of 1Gbps. A 5µs inter-transmission guard interval is used, while the ONU warm-up overhead is 2ms. The cycle duration is varied from 10 to 90ms and the performance is mainly studied under 50% and 90% loads. The rest of the parameters are summarized in Table 5.1 along with their description and default values.

5.1.4.1 General Energy Performance

Figure 5.3 shows the energy-saving percentages of all schemes in relation to the chosen cycle duration. The figure shows three sets of graphs representing network loads of 10, 50, and 90%. It is clear how energy-saving gains increase with increasing the cycle duration as well as under light loads. For the presented schemes, there is no significant variation in their performance under light loads. Under heavy loads, however, *fixed-slot* (FS) and *fixed-slot with full-credit* (FS-FC) have the highest energy savings, whereas gap-filling schemes have the lowest savings.

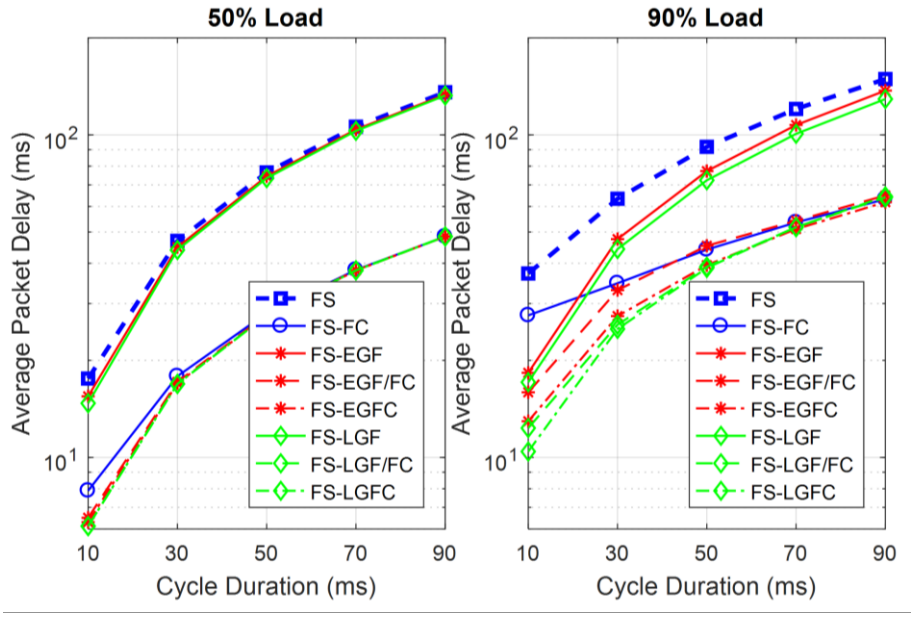


Figure 5.4: Delays vs. cycle duration.

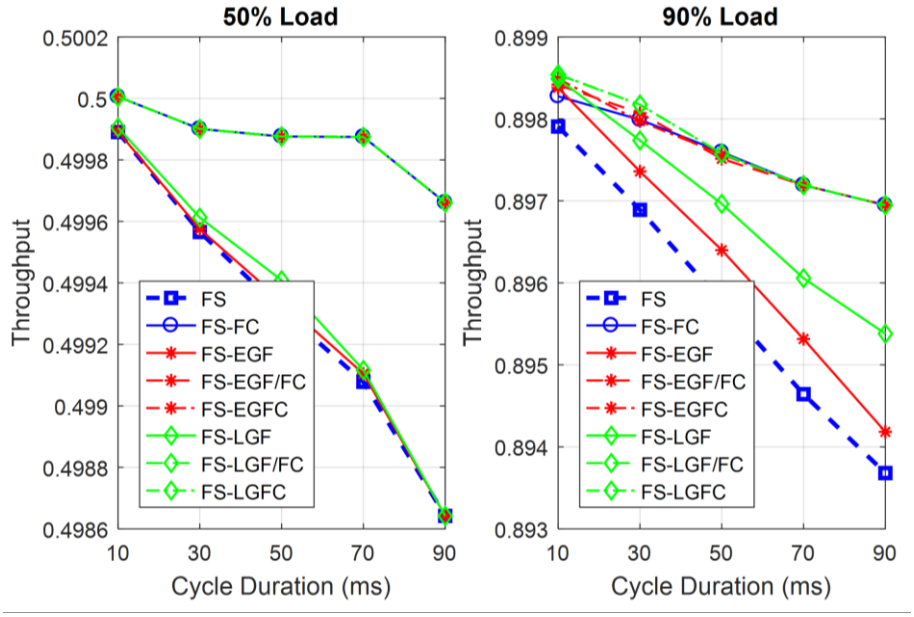


Figure 5.5: Throughput vs. cycle duration.

5.1.4.2 Delay/Throughput Tradeoff with Cycle Duration

Figures 5.4 and 5.5 show the effect of extending the cycle duration on the network performance in terms of delay and throughput, respectively. While using longer cycles enhances energy savings, it also degrades the network performance. Pure *fixed-slot* (FS) allocation clearly

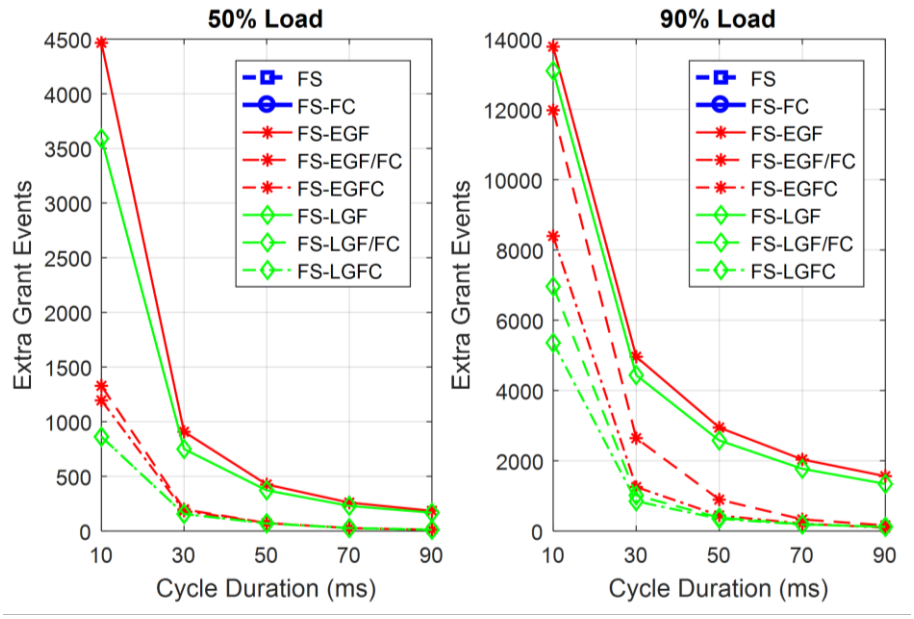


Figure 5.6: Extra grant counts for gap-filling schemes.

has the worst performance, whereas using it with *full-credit* (FS-FC) significantly improves its performance and holds a middle ground between the gap-filling schemes. Pure gap-filling schemes have a performance falling between FS and FS-FC at cycles greater than 10ms while combining gap-filling with credit appears to achieve the overall best network performance. The best scheme is found to be *largest gap-filling with credit* (LGFC), which is shown to have the best upstream utilization since it helps relieve heavily-loaded ONUs with the largest available gaps.

5.1.4.3 Extra Grant Counts (Gap Filling Events)

Figure 5.6 compares between the gap-filling schemes in terms of their gap-filling events. Using credit together with gap-filling (the lower 4 graphs) greatly decreases the gap-filling count since many packets will already be granted transmission within the normal allocated credit. Moreover, both largest-gap-filling schemes (dotted green graphs), namely LGF/FC and LGFC, have the lowest counts since their extra windows are larger and carry more traffic. This also translates in their power-usage since ONUs do not stay awake as frequent as in earliest gap-filling.

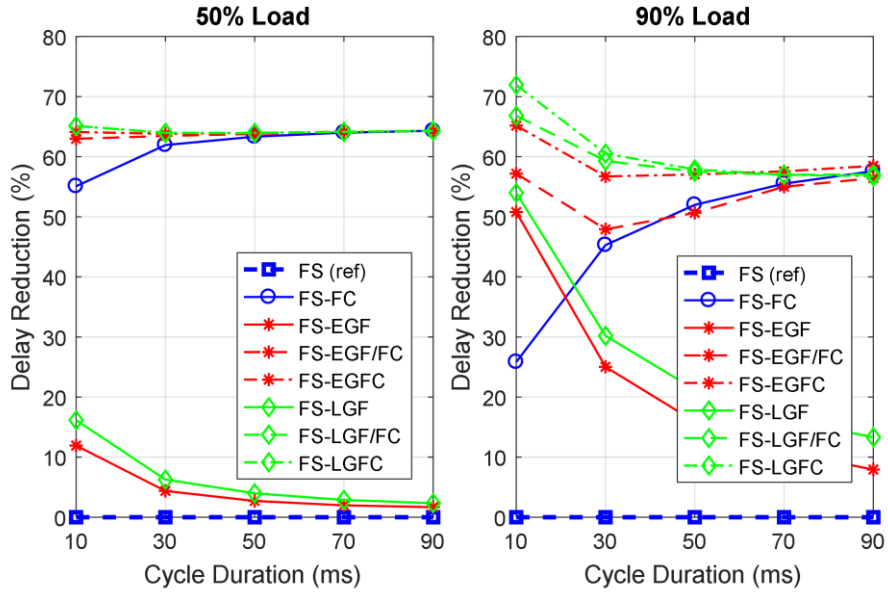


Figure 5.7: Normalized delays vs. cycle durations.

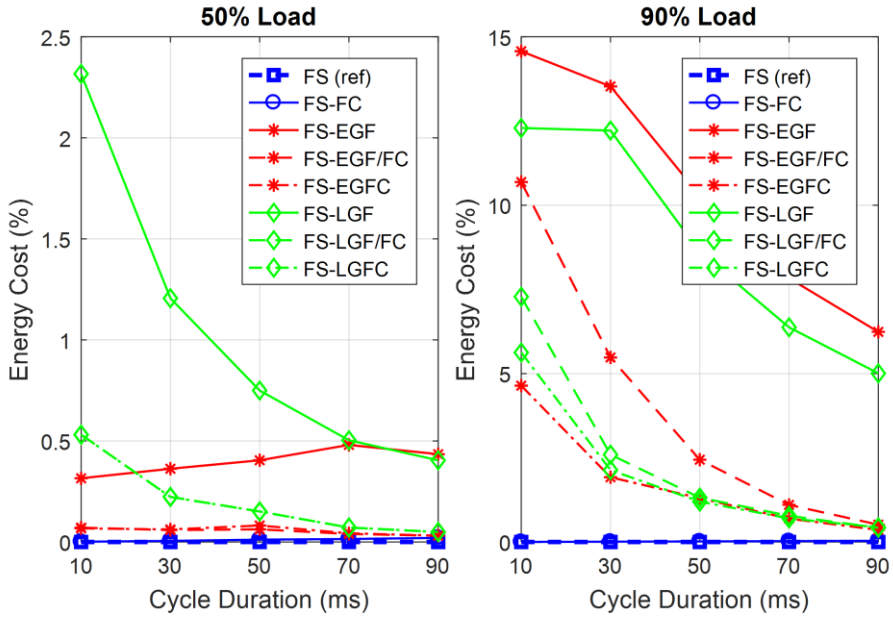


Figure 5.8: Power cost vs. cycle durations.

5.1.4.4 Delay-Power Tradeoff

To better visualize the associated tradeoffs with improving the network performance, Figures 5.7 and 5.8 show the delay reductions (improvement in delays compared to FS) and the corresponding energy cost (additional power consumption) of all schemes. Using credit with gap-

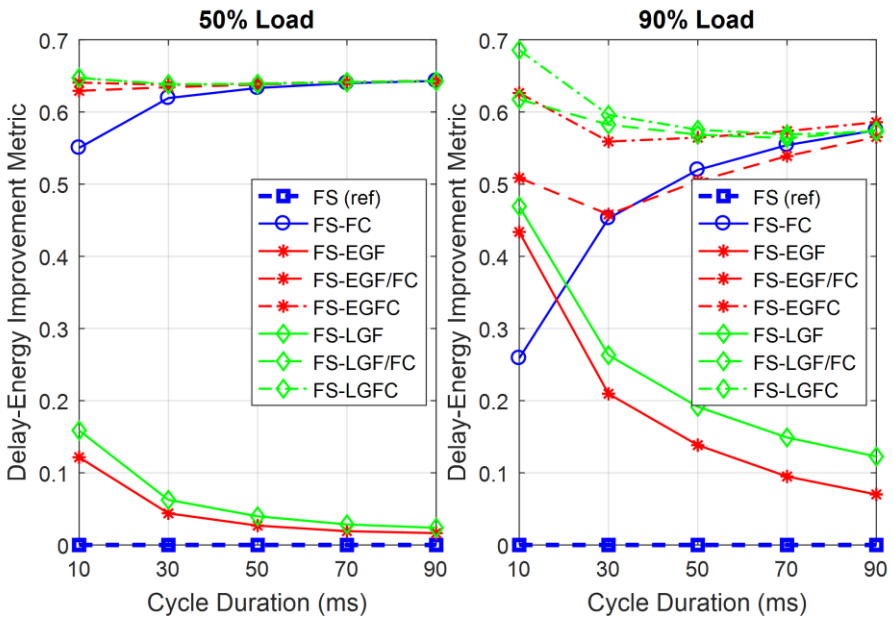


Figure 5.9: Improvement index vs. cycle durations.

filling again appears to have the best performance in terms of delay with almost 60% reduction in delays. Using credit alone does not perform as well, but is still better than only using gap-filling, especially under light loads. On the other hand, fixed-slot and full-credit have the least energy cost. Still, the additional power required for schemes using gap-filling with credit is not significant and does not exceed 0.5% under light loads nor 5% under heavy loads. EGF is shown to require less energy than its LGF counterpart, especially under light loads, whereas using gap-filling alone shows to have the highest energy consumption.

5.1.4.5 Improvement Index

To determine which of the schemes has the overall best performance, considering both delay and power metrics, we compute the improvement index for each scheme following the expression given by Eq. (5.2). Figure 5.9 shows the computed improvement index with fixed-slot allocation being the reference scheme with an index of 0. The overall best scheme appears to be LGFC with a close match with LGF/FC. Earliest-gap-filling schemes, however, seem to

outperform their largest-gap-filling counterparts at cycle durations above 70ms. This because at such long cycles, early windows become vital for lower packet delays and less awake times compared to short cycles where larger windows would then be more useful.

5.2 Proposed Decentralized Energy-Conserving Framework

In this part of the chapter, we turn our attention to energy-efficiency within decentralized allocation. Because ONUs need to communicate together to manage media access, the requirement for energy efficiency poses new and unique challenges in decentralized allocation. This is because placing additional transceivers within ONUs for inter-ONU communications tends to add to their power consumption. Moreover, the ONUs' continuous management of upstream media access may interfere with putting them to sleep. In this section, we address these challenges and propose a novel decentralized-based bandwidth allocation framework, which is designed to succeed on three main fronts; achieving high network performance, conserving energy, and supporting faster cloudlet offloading.

5.2.1 The Additional Transceiver

In previous chapters, we have seen how decentralized allocation has better network and offloading performances than centralized allocation. These performance gains come at the cost of using additional transceivers to acquire inter-ONU communications. This, however, implies more power consumption, which adds to the operational cost of the network. In this section, we explore how this additional power cost can be minimized by closely examining the inner workings of the ONU's main upstream transceiver.

Because ONUs are the most power-hungry devices, minimizing their energy consumption is typically done by minimizing their awake times and maximizing their sleep times. Another

important factor for saving energy is the amount of power consumed during active and sleep modes. Assuming that the cycle C is equally divided between N ONUs and that the wake-up overhead T_{oh} is much smaller than C ($T_{oh} << C$), the energy consumption of the ONU's main transceiver can be expressed as:

$$E = P_a(\frac{1}{N}C + T_{oh}) + P_s(\frac{N-1}{N}C - T_{oh}) \tag{5.3}$$

where P_a and P_s are the ONU's active and sleep power consumptions, respectively. The percentage of energy saved when comparing to an ONU that is always awake is then:

$$\eta = \frac{P_aC - E}{P_aC} = \left(\frac{N-1}{N} - \frac{T_{oh}}{C}\right)\left(1 - \frac{P_s}{P_a}\right) \tag{5.4}$$

The expression above stresses two fundamental ratios for TDM PON power savings; a time ratio and a power ratio [71]. The time ratio T_{oh}/C indicates that minimizing the wake-up recovery overhead would improve energy savings. On the other hand, the power ratio P_s/P_a highlights another important factor, which is the ONU's power consumption in its idle state (P_s) that dominates nearly (N–1)/N of the cycle, as seen in Eq. (5.3). Therefore, minimizing both the recovery time T_{oh} and idle power P_s would significantly improve energy efficiency [71].

To improve the clock recovery overhead T_{oh}, some studies suggested keeping parts of the ONU's receiver powered on during sleep mode, such as the *clock and data recovery* (CDR) circuit, in order to maintain OLT synchronization [39]. This however presented a tradeoff between the overhead T_{oh} and the power incurred during sleep mode P_s. This tradeoff is demonstrated in Table 5.2, which lists wake-up overheads and corresponding power consumption of three different receiver settings when the front-end analog circuit is completely or partially switched off [39]. Incorporating a *burst-mode* (BM) CDR was shown to incur less power than the default *continuous mode* (CM) circuit, which requires more components to be left on for synchronization.

Table 5.2 ONU Receiver Parameters in the Literature [39]

	Receiver Block (*including Backend Digital Circuit*)		
	Setting 1 (default)	***Setting 2 (BM-CDR)***	***Setting 3 (CM-CDR)***
Active Power (W)	3.85	3.85	3.85
Sleep Power (W)	0.75	1.08	1.28
Total Overhead	2 – 5.125ms	125µs	125µs
Clock Recovery	2 – 5ms	1 – 10ns	0
Synchronization	125µs	125µs	125µs

Table 5.3 ONU Transmitter Parameters in the Literature

	Transmitter Block		
	DFB [79], [81]	***LC-VCSEL*** [68], [69]	***SC-VCSEL*** [69]
Active Power (W)	1.202	0.134	0.1
Settling Time (ns)	760	330	205

Table 5.4 Power Consumption in each ONU Mode

	Whole ONU Transceiver		
	DFB [79], [81]	***LC-VCSEL*** [68], [69]	***SC-VCSEL*** [69]
Active Power (W)	5.052	3.984	3.95
Dozing Power (W)	3.85	3.85	3.85
Sleep Power (W)	0.75, 1.08, or 1.28[α]		

[α] The sleep power depends on the chosen receiver block setting [39].

Other studies focused on reducing the active power consumption P_a itself to improve energy efficiency, especially in cases where ONU idle times are not long enough to allow them to switch to sleep mode. For instance, replacing the *distributed feedback laser* (DFB) with a *long-cavity vertical-cavity surface-emitting laser* (LC-VCSEL) was found to reduce active power consumption by an order of magnitude [68]. The power can be further reduced by using a *short-cavity* (SC) laser, which also shortens the transmitter's settling time [69]. These three types of transmitters are listed

in Table 5.3, which shows both their power and settling time. The total transceiver power then depends on the choice of receiver setting and transmitter type, as summarized in Table 5.4.

Now, to employ an additional ONU transceiver for OOB communications in a decentralized scheme, a few differences need to be taken into consideration. The first is having no OLT clock recovery on the additional channel. Instead, the periodic clock recovery achieved by the primary receiver can be sufficient for synchronizing the ONUs' local clocks for inter-ONU communication. Note also that the transmission of a single tag frame on an OOB channel with a rate of 1Gbps is very short (~576ns) compared to the guard interval used (5µs), which allows some degree of ONU clock variation. We therefore adopt the default receiver setting in Table 5.2 for the additional receiver, except that it will only have a warm-up overhead of 125µs. Moreover, both receivers can share a single backend digital circuit containing core ONU components that cannot be switched off, such as the local clock and volatile memory. That way, the incremental power consumption due to using the additional transceiver would only be incurred during its active and doze states.

5.2.2 Signaling Ahead with Dense Tagging (DT)

In previous chapters, the term 'tagging' has been used to refer to the exchange of control messages between ONUs on the OOB channel. Figure 5.10 illustrates the tagging schemes that we use in this chapter to compare the performance of our proposed decentralized scheme. The first one belongs to TTACT, where the tag frame informs the following ONU of the duration and start time of the current transmission. Accounting for propagation delays and a guard interval, the following ONU would be able to calculate its time of upstream media access. Therefore, in TTACT, only one ONU may be sending a tag, while all other ONUs must listen to know when to access the media. Because the time of being tagged is unknown, an ONU must continuously wait for its preceding

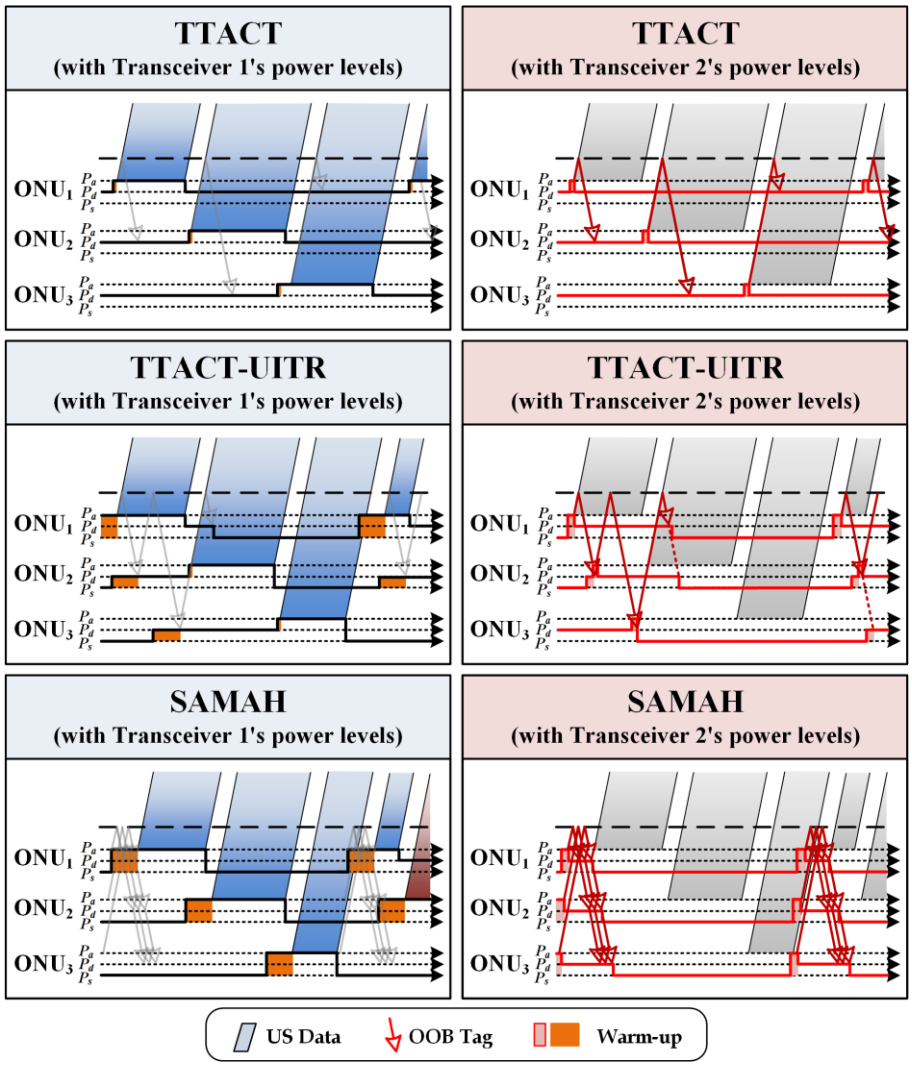


Figure 5.10: Transceivers power levels for different tagging schemes.

ONU's tag and be ready to access the media soon after, which means that it cannot switch off either one of its transceivers, but can only stay in doze mode.

The second tagging scheme is the modification we proposed in Chapter 4, which we called *unconditional immediate tagging with reset* (UITR). In UITR, the ONUs are forced to exchange tags at the beginning of each cycle, reporting only the packets buffered up until that point. This decouples the time of sending a tag frame from the time of upstream transmission and, by requiring the first ONU in the sequence to wait until the next cycle before restarting this tagging process,

it creates a silent OOB period from the time tags are exchanged until the start of the next cycle. This period is then used for exchanging edge traffic between ONUs. Because tags are immediately exchanged at the beginning of each cycle, each ONU can use the information gathered from all tags to compute the start time of the next cycle. This allows it to switch its secondary transceiver to sleep mode and wake up before the next tag exchange. On the other hand, the main transceiver must be awake at the beginning of each cycle since an ONU may be granted access shortly after the tag exchange if some or all preceding ONUs had nothing to transmit. It can then be switched off after media access or after tag exchange.

Finally, for our proposed decentralized framework in this chapter, we use what we call *dense tagging* (DT), which allows all tags to be sent during a tag exchange period. However, with the ONUs accounting for propagation delays and tag frame alignment, the tags are received at each ONU as if they were sent one after another in a very short window. Because the ONUs exchange tags and schedule their media access before the cycle starts, we call the framework *Signaling-Ahead for Media Access Handovers* (SAMAH). Exchanging tags ahead of the cycle enables the ONUs not only to know when the next tag exchange will take place but also know when they will access the media, which allows turning off both transceivers for long durations. However, this early knowledge may not reflect the actual bandwidth demand at the time of media access. Instead, it may lead to sending only a fraction of queued upstream traffic and delay the rest for the next cycle. To overcome this, several techniques are proposed in the following section.

5.2.3 Proposed Upstream Bandwidth Allocation

Decentralized media access in TTACT is online-based, in which the duration of an ONU's transmission window W_i is solely determined by its current buffer queue size B_i and its maximum allowable window $W_{i,\max}$:

Table 5.5 Credit-Based Allocation Techniques

	Formula
Ratio Credit (RC)	$c_i = W_i(W_{i,\max} - B_i)/W_{i,\max}$
Sequence Credit (SC)	$c_i = i(W_{i,\max} - B_i)/N$
SC+RC (SRC)	$c_i = (i/N + B_i/W_{i,\max})(W_{i,\max} - B_i)$
Linear Credit (LC)	$c_i = aB_i, \quad a > 0$
Constant Credit (CC)	$c_i = bW_{i,\max}, \quad b < 1$
Full Credit (FC)	$c_i = W_{i,\max} - B_i$
Estimated Credit (EC)	$c_i = \overline{(\Delta B)_i}$

$$W_i = \min(B_i, W_{i,\max}) \tag{5.5}$$

In other words, the ONU does not have nor require knowledge of the bandwidth requirements of other ONUs. On the contrary, tags in SAMAH are exchanged ahead of time, giving ONUs full knowledge of each other's requirements before the cycle starts. This, however, means that the ONUs may be using premature information that does not reflect their actual requirements at the time of media access to set their transmission windows. In the following, we examine several approaches that aim to mitigate this issue.

5.2.3.1 Credit-Based Approach

One way to alleviate the premature reporting problem within SAMAH is to allow ONUs to allocate themselves additional credit. This may help them accommodate newly arriving packets after tag exchange. An ONU would then set and announce its transmission window according to:

$$W_i = \min(B_i + c_i, W_{i,\max}) \tag{5.6}$$

where c_i is some extra bandwidth credit that reflects an expectancy of newly arriving traffic. Table 5.5 summarizes different methods of computing c_i. *Constant* and *linear* credits are not expected to differentiate between ONUs based on how early their buffer reports are, whereas other

more advanced techniques allocate credit based on the ONU's place in the transmission sequence or its average buffer variance. Note that W_i is still upper bounded by $W_{i,\max}$ and that *full credit* (FC) always meets this upper bound. This is why credit-based schemes are unable to aid heavily-loaded ONUs already fully utilizing their maximum windows.

5.2.3.2 Excess Distribution (ED) Approach

Excess distribution relaxes the upper limit $W_{i,\max}$ for heavily-loaded ONUs, without exceeding the cycle duration, by allowing them to use excess bandwidth from lightly-loaded ONUs. This would typically require an offline-based allocation framework, in which bandwidth requirements of all ONUs must be known before distributing excess bandwidth among heavily-loaded ONUs. In centralized allocation, such a framework would normally result in an idle period (i.e., a walk time) of roughly one RTT, from the time of sending the last ONU report to the OLT until the first grant is received [82]. One way of mitigating this is to use a hybrid framework, where lightly-loaded ONUs are immediately granted bandwidth (online) to utilize some of the idle time, whereas overloaded ones are delayed until all reports are gathered (offline) so that a more informed decision can be made [89].

In the proposed SAMAH framework, excess distribution is carried out offline without incurring any idle periods since the ONUs exchange information ahead of the cycle on an auxiliary channel. SAMAH therefore unlocks the full potential of having global network information without suffering any of the drawbacks associated with centralized offline scheduling. Moreover, contrary to the OLT's decision making in centralized allocation, each ONU in SAMAH makes its own decision using the information gathered from the other ONUs. The ONUs then do not need to announce their decisions afterward since they can easily deduce each other's decisions and arrive at the same outcome independently.

To perform excess distribution in SAMAH, ONUs first report their queue sizes B_i in their outgoing tags instead of reporting their self-allocated transmission windows W_i as given by Eq. (5.5). Under overloaded network conditions ($\Sigma B_i > NW_{i,\max}$), ONUs are partitioned into two groups according to their reports: lightly-loaded ($B_i \leq W_{i,\max}$) and heavily-loaded ($B_i > W_{i,\max}$). Each ONU can then compute the overall cycle excess from the set L of lightly-loaded ONUs as:

$$C_{excess} = \sum_{i \in L} (W_{i,\max} - B_i) \tag{5.7}$$

After that, each ONU computes its own window, without needing to announce it, as:

$$W_i = \begin{cases} B_i, & B_i < W_{i,\max} \\ W_{i,\max} + \dfrac{(B_i - W_{i,\max})C_{excess}}{\sum_{j \notin L} (B_j - W_{j,\max})}, & B_i \geq W_{i,\max} \end{cases} \tag{5.8}$$

Note that the excess distribution proposed here is overload-driven, in which excess is distributed among heavily-loaded ONUs according to their overload [82]. Under normal network conditions ($\Sigma B_i < NW_{i,\max}$), the excess is instead distributed among all ONUs according to their demand:

$$W_i = B_{i,\max} + B_i C_{excess} \Big/ \sum_{j=1}^{N} B_j \tag{5.9}$$

After each ONU determines the size of its upcoming window, it would also need to compute the windows for the preceding $i - 1$ ONUs in order to schedule its upstream media access subsequently, leaving $i - 1$ guard intervals.

5.2.3.3 **Hybrid Approach**

Even though excess distribution improves statistical multiplexing, it assumes that all queue reports are accurate and up-to-date, which is not the case in SAMAH. Credit-based techniques, on the other hand, do not offer any advantages to heavily-loaded ONUs already utilizing their maximum bandwidth. We therefore propose a hybrid approach that aims to capture the best of each

Table 5.6 Credit Distribution Allocation Techniques

	Formula
RC Distribution (RCD)	$W_i = B_i + B_i C_{excess} \Big/ \sum_{j=1}^{N} B_j$
SC Distribution (SCD)	$W_i = B_i + iC_{excess} / (N(N+1)/2)$
EC Distribution (ECD)	$W_i = B_i + (B_i + \overline{\Delta B_i}) C_{excess} \Big/ \sum_{j=1}^{N} (B_j + \overline{\Delta B_j})$

approach; allowing heavily-loaded ONUs to surpass their maximum limit while also accounting for untimely reports of later ONUs. In the following, we introduce novel hybrid schemes that can be realized with SAMAH's offline-based framework.

Posterior Excess Distribution:

In this hybrid approach, each ONU first performs a credit-based pre-allocation as given by Eq. (5.6). Afterward, all ONUs report their computed windows in their outgoing tags before performing excess distribution using their credit allocated windows as the basis for its calculations. This tilts excess distribution to some degree in favor of a selected credit-based allocation scheme. For this hybrid approach, we combine *excess distribution* with; *ratio credit* (RC), *sequence credit* (SC), or *sequence and ratio credits* (SRC), since they differentiate between ONUs based on their network loads and place in the transmission sequence.

Credit Distribution:

In this hybrid approach, the ONUs do not perform any initial pre-allocation. Instead, they only report their queue sizes B_i in their outgoing tags, as done in normal excess distribution. The ONUs then calculate the excess and distribute it according to one of the methods of Table 5.6.

Note that, in the estimation-driven case, each ONU must additionally report its buffer expectancy $(B_i + \overline{\Delta B_i})$ by computing the average buffer variation over past cycles from the time of sending its tag until the time of media access. This approach therefore empowers credit-based allocation by distributing all cycle excess as credit and may extend an ONU's window beyond the $W_{i,\max}$ limit.

5.2.4 Network Operation and DT Initialization

Similar to TTACT, the ONUs manage upstream media access as well as OOB offloading in SAMAH. Although upstream bandwidth allocation is decentralized, it is still governed by the OLT. The OLT first initializes bandwidth allocation by setting the ONUs' transmission sequence along with the maximum window of each ONU according to its *service level agreement* (SLA). It additionally sets the chosen media access scheme and cycle configuration whether it is adaptive or fixed. The OLT therefore still maintains control over the network making the ONUs only manage their upstream media access based on the initial control parameters they receive. The OLT also carries out most of its other functionalities such as ONU auto-discovery. The ONUs, on the other hand, do not have to wait for grants to transmit since they operate based on their preset values. That way, the OLT only needs to send bandwidth allocation settings whenever a parameter is required to change, whereas other parameters that are changed within each cycle, such as each ONU's window size and its timing, are locally managed between ONUs.

Figure 5.11 shows the initialization phase of SAMAH's dense tagging along with the power levels of each transceiver. During SAMAH's initialization, the OLT must additionally define an initialization period, after which the first cycle starts. This period accounts for the ONUs' OOB propagational delays, tag frame alignment, and processing times. In the worst case (if all ONUs were at equal distances from the OLT), this period will be no longer than:

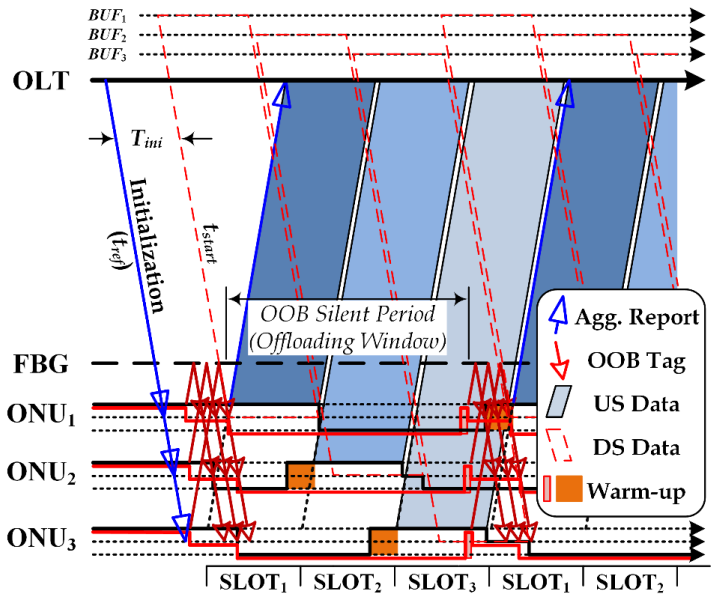


Figure 5.11: Initialization of SAMAH's dense tagging.

$$T_{ini} \le \max_{\forall i}(2D_{i,SC}^{prop} + (N-i+1)(T_{tag}+T_g) + 2D_{ONU}^{proc}) \tag{5.10}$$

where $D_{i,SC}^{prop}$ and D_{ONU}^{proc} are ONU$_i$'s propagation delay to the splitter and its processing delay, respectively, whereas T_{tag} is the transmission delay of a tag frame. Using a common time reference t_{ref}, the ONUs can compute the start time of the next cycle as:

$$t_{start} = t_{ref} + T_{ini}^{\max} + D_{ONU}^{proc} \tag{5.11}$$

where t_{ref} can either be explicitly specified by the OLT (after synchronizing to its timestamp) or it can be the time of receiving the broadcasted initialization frame by each ONU. After the cycle start is known, each ONU computes the time at which it sends its tag frame as:

$$t_i^{TAG} = t_{start} - 2D_{i,SC}^{prop} - (N-i+1)(T_g+T_{tag}) - D_{ONU}^{proc} \tag{5.12}$$

The cycle itself can be either dynamic or fixed depending on the chosen setting:

$$C = \begin{cases} \sum_{i=1}^{N} W_i + NT_g, & \text{if adaptive cycle} \\ C_{\max}, & \text{if fixed cycle} \end{cases} \tag{5.13}$$

Table 5.7 Symbol Definitions and Simulation Values

Symbol	Description	Value
$L_{\max}$	Maximum network span	100km
$L_{i,SC}$	ONU$_i$'s distance from splitter/combiner	0–5km
L_{SC}	Splitter distance from OLT (feeder span)	95km
R_{down}	Downstream transmission rate	10Gbps
R_{up}	Upstream transmission rate	1Gbps
R_{OOB}	Out-of-band channel data rate	1Gbps
R_{user}	Connected users' upstream rate	100Mbps
N	Number of ONUs	16
Q	ONU buffer size	10MB
L	Normalized network load	0.5, 0.9
C	Cycle duration	10 – 90ms
T_g	Guard interval	5µs
T_{oh}	Warm-up overhead	2ms (DFB), 0.125ms (SC)
T_{set}	Doze overhead	760ns (DFB), 205ns (SC)
P_a	ONU power when fully awake	5.052W (DFB), 3.95W (SC)
P_d	ONU power during doze mode	3.85W (DFB), 3.85W (SC)
P_s	ONU power during sleep mode	0.75W

where $C_{\max}$ is a fixed value defined by the OLT, at which the cycle is kept regardless of the sizes of upstream transmissions to allow for sufficient ONU sleep durations. SAMAH is shown with a fixed cycle setting in Figure 5.11, where each ONU is shown to fully use its slot in both transmission directions for illustrative purposes. Once an ONU finishes its upstream transmissions, it switches to doze mode until it receives all its downstream frames. Moreover, in order to maximize downstream and upstream overlap, the first ONU in the sequence must send an aggregated report to the OLT, at the beginning of each cycle, to help the OLT track ONU awake times and schedule downstream transmissions accordingly. This maximizes sleep times by making OLT transmissions adapt to varying upstream traffic from one cycle to another.

5.2.5 Numerical Results for the Decentralized Energy-Saving Framework

For our decentralized framework, the upstream transceiver is modeled as a DFB similar to [79] and [81]. For the secondary transceiver, we model it as either DFB or SC-VCSEL, since

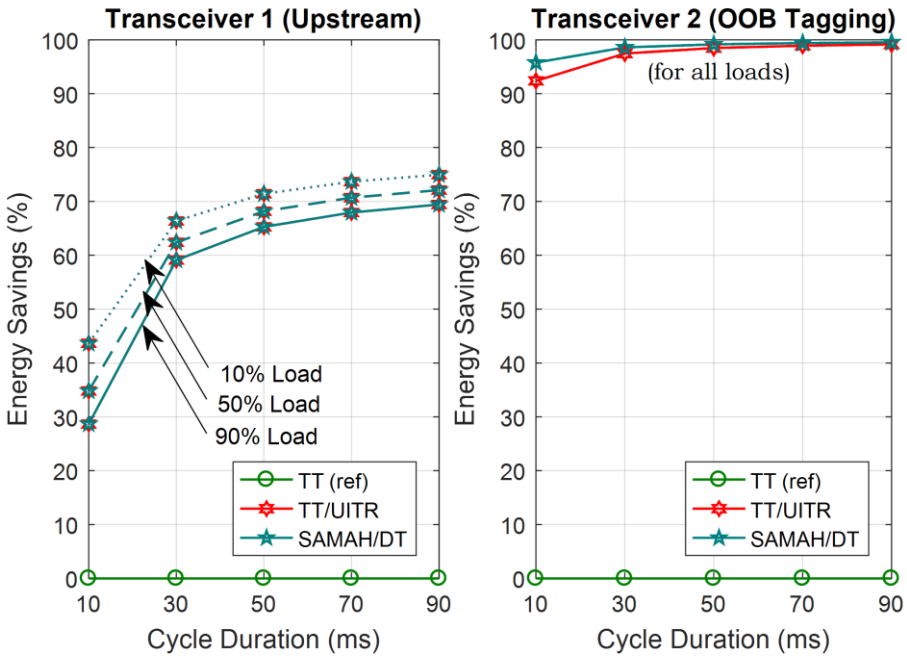


Figure 5.12: TT/UITR and SAMAH energy savings compared against TT.

the energy consumption of the SC-VCSEL and LC-VCSEL transmitters are almost the same. Both ONU transceivers share a backend digital circuit and the cycle duration is varied from 10 to 90ms. Because we use a fixed cycle setting, we remove the *adaptive cycle time* (ACT) acronym from TTACT from here on out. The rest of the simulation parameters are listed in Table 5.7.

5.2.5.1 Energy Performance of SAMAH with DT

Figure 5.12 shows the energy-saving percentages achieved by both UITR and SAMAH's *dense tagging* (DT) when compared against the default tagging scheme of *taking turns* (TT). For the main transceiver, UITR and DT have very similar energy savings going up to 70% with longer cycle durations since they both enable ONUs to sleep, unlike TT's default tagging scheme. On the other hand, savings on the secondary transceiver are much higher (above 90%) especially for SAMAH, which enables slightly longer sleep durations of 95.8%, 99.2%, and 99.5% compared to 92.4%, 98.5%, and 99.2% of UITR at 10, 50, and 90ms, respectively. Note that savings here are

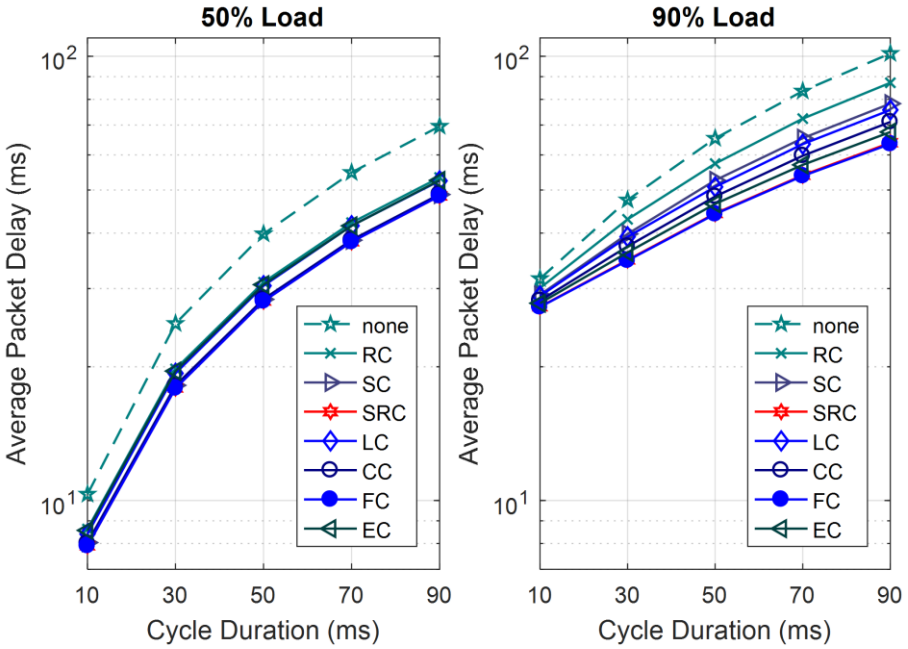


Figure 5.13: Upstream packet delays of credit-based schemes.

independent of the network load since the secondary transceiver is only used for tag exchange. To compare the delay performance of these tagging schemes, we first evaluate the different approaches that aim to overcome SAMAH's early reporting drawback.

5.2.5.2 Credit-Based Approach

Figure 5.13 shows how SAMAH's upstream packet delays can be reduced when credit-based schemes are applied. The corresponding delay reduction is shown in Figure 5.14, where all credit-based schemes succeed in bringing down the delays of SAMAH's pure DT with the best being *full credit* (FC). Using *sequence and ratio credits* (SRC) almost has the same performance of FC indicating that combining both credits often allows ONUs to use their maximum bandwidth given in FC.

Figure 5.15 shows the energy performance of the credit-based schemes compared to SAMAH with no credit, considering the consumption of both transceivers. The variation in energy

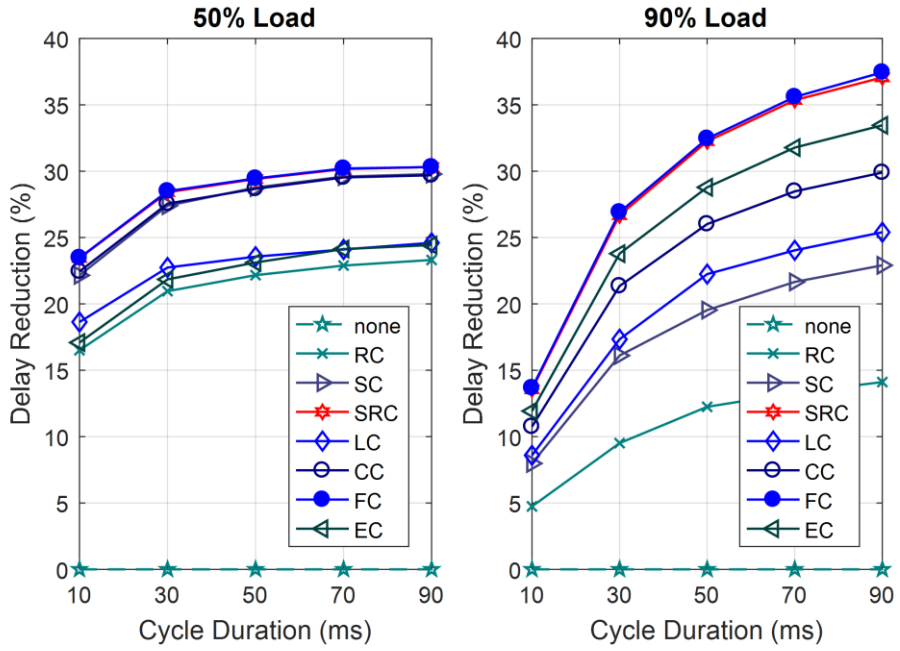


Figure 5.14: Delay reduction of credit-based schemes.

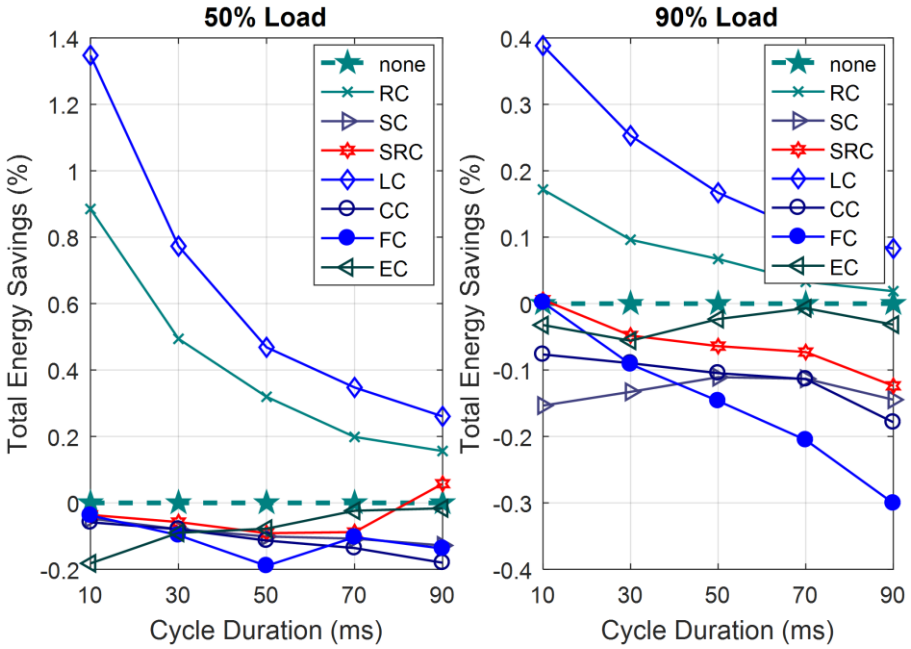


Figure 5.15: Energy savings of credit-based schemes.

performance at 90% loads is less than ±0.5% and ±1.5% for 50% and 90% loads, respectively, where *linear credit* (LC) and *ratio credit* (RC) are found to be the only credit schemes that have better energy performance.

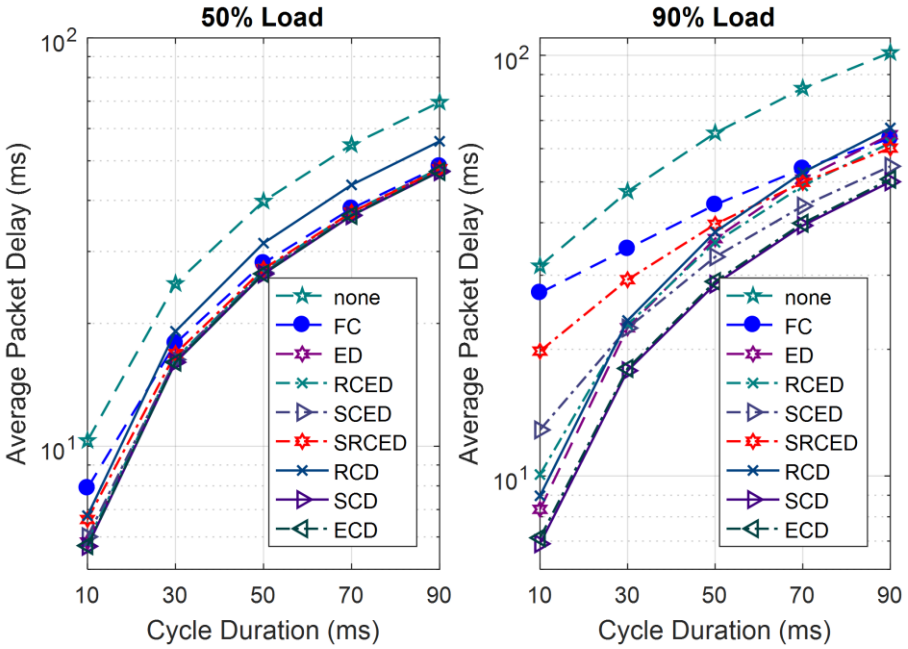


Figure 5.16: Upstream packet delays of excess/hybrid schemes.

5.2.5.3 Excess and Hybrid Distribution Approaches

Figure 5.16 shows the delays of the excess distribution and hybrid distribution schemes. FC is also shown for the sake of comparison since it is the best among credit-based schemes. The corresponding delay reduction is shown in Figure 5.17, where most excess distribution schemes succeed in bringing down SAMAH's delays even beyond that achievable with FC. Pure *excess distribution* (ED) is shown to have better performances then *ratio-credit distribution* (RCD), which proves that an overload-driven excess distribution in ED is more efficient than the conventional demand-driven distribution used for RCD. On the other hand, *sequence credit distribution* (SCD) and *estimated credit distribution* (ECD) appear to have the overall best performance regardless of the cycle duration, reducing the delays by more than 50% under heavy loads. This is because these two schemes greatly account for the ONUs' place in the transmission sequence and employ all the cycle excess to compensate for early reports.

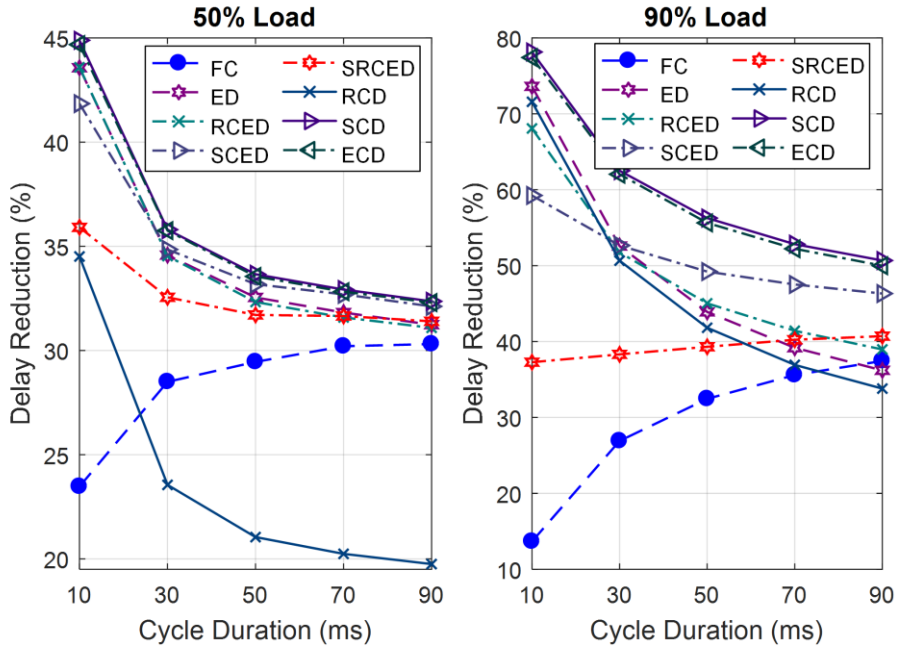


Figure 5.17: Delay reduction of excess/hybrid schemes.

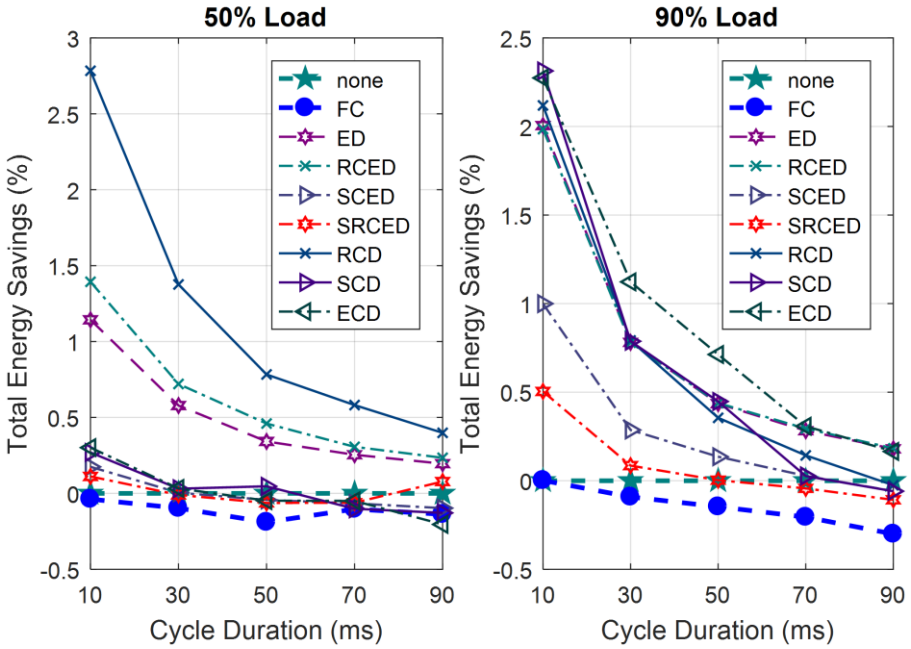


Figure 5.18: Energy savings of excess/hybrid schemes.

Figure 5.18 shows the energy performance of these excess and hybrid schemes. The additional energy savings are less than 3% and decrease as the cycle duration is extended. With short cycles, most schemes are found to have lower power consumption than that of pure DT.

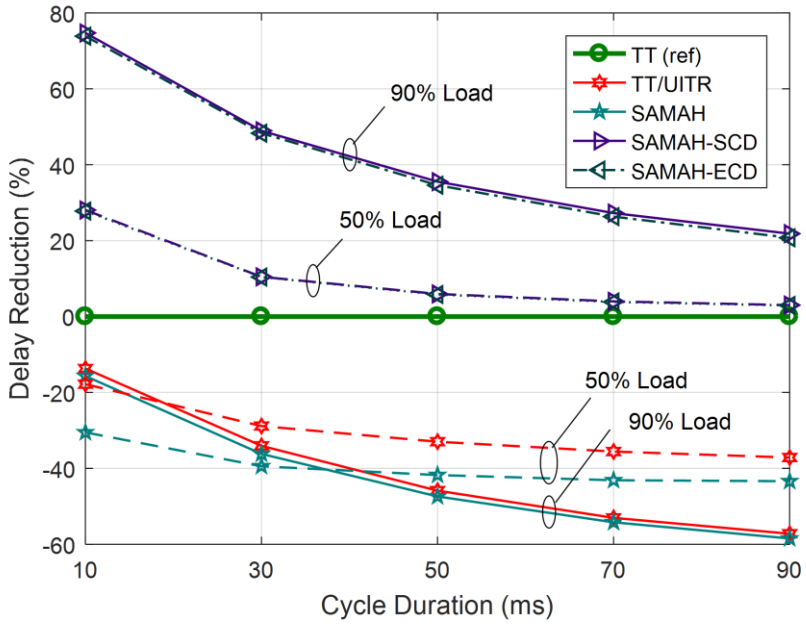


Figure 5.19: Delays of different tagging with SAMAH's enhancements.

Again, FC is shown for the sake of comparison, where it can be seen how excess and hybrid schemes have lower delays and better energy performance than credit-based schemes. The two most promising schemes are found to be SCD and ECD, which we use next in our delay performance comparison against TT and UITR.

5.2.5.4 Delay Performance of SAMAH with DT

Figure 5.19 compares the delay performance of the different tagging schemes. Even though SAMAH with pure dense tagging was found to be the most energy-efficient when compared against other tagging schemes (Figure 5.12), it also happens to have the longest delays due to its premature reports. However, when applying SCD and ECD to enhance its bandwidth allocation, SAMAH then achieves significant delay reductions, enabling it to have the lowest delays among all decentralized schemes.

5.3 Conclusions

In the first part of this chapter, we evaluated different centralized schemes as part of an energy conservation framework for long-reach optical access networks. The schemes aimed to improve the network performance of fixed-slot allocation, which lacks the dynamicity of conventional bandwidth allocation schemes and does not give heavily-loaded ONU any special treatment. Numerical results have shown that, with a small additional power cost (less than 5%), packet delays can be reduced by nearly 60% through combining gap-filling with credit-based schemes. Moreover, the power savings due to extending the cycle duration longer than 50ms appear to be insignificant compared to the accompanying network performance degradation. In the second part of the chapter, we proposed a decentralized energy conservation framework and compared it against its decentralized predecessors. Numerical results showed that the proposed decentralized framework (with its applied enhancements) has the overall best performance in terms of both packet delays and energy efficiency.

Chapter 6

Fog-LR-PON Integration:

an Energy-Efficiency Perspective

In the previous chapter, we developed power-conserving frameworks for both centralized and decentralized-based allocation, in which their bandwidth allocation is aware of the cyclic sleep mode that the *optical network units* (ONUs) go through to conserve energy. In this chapter, we reexamine the fog integration problem using these energy-efficient frameworks as the underlying bandwidth allocation schemes. We study and compare the trade-offs of each allocation paradigm to determine which has the overall performance, mainly in terms of energy-efficiency and offloading delays.

6.1 Network Architecture and Cloudlet Placement Revisited

The first design consideration for integrating fog resources with optical access networks is the optimal placement of the fog cloudlets. They need to be as close as possible to end devices without requiring expensive architectural modifications or excessive fiber deployments. In Chapter 3, we studied the three common cloudlet placements in optical access networks, which are near the *central office* (CO), at the *remote nodes* (RNs), or in the field close to the ONUs. We have seen how some placements require fewer cloudlet deployments and how others offer

Table 6.1 Common Cloudlet Placements in a Typical PON

	OLT Cloudlet (Metro Cloudlet)	RN Cloudlet (Access Cloudlet)	ONU Cloudlet (Edge Cloudlet)
Accessibility	shared by OLTs (many networks)	shared by ONUs (same network)	dedicated for each ONU
Resources	vast, low rejection ratio	medium, average rejection ratio	limited but dedicated, low rejection ratio
Cost/Deployment	powerful cloudlet/ for each CO	moderate cloudlet + RN switching/ for each RN	light-weight cloudlet + ONU connections/ for each ONU
Delay	long	not long	minimal
Tasks	storage/ delay-tolerant	delay-sensitive	real-time

lower latencies. Table 6.1 highlights the characteristics of each cloudlet placement and its potential tradeoffs. Note that if a cloudlet is shared or has limited resources, a service rejection ratio is introduced corresponding to the probability of its resources being occupied. Note also that a hybrid architecture may be used that employs different types of cloudlets (see for example [67], [83]).

In previous chapters, we have also seen how the decentralized architectural modification enables ONU-cloudlets to be easily shared among other ONUs through *out-of-band* (OOB) inter-ONU communications. This could support an uneven ONU-cloudlet distribution, where cloudlets may only be connected to a subset of ONUs that are serving the largest number of constrained devices but still be accessible by other ONUs. In this chapter, we expand upon this possibility further and propose to employ a shared access cloudlet that does not require active RN filtering. Instead, the cloudlet can be connected to the RN with a fiber connection the same way an ONU is connected. The same inter-ONU communications proposed for decentralized allocation would then allow the ONUs to directly communicate with this common access cloudlet. This also allows for cases where the access cloudlet may, in fact, be an advanced ONU with abundant resources.

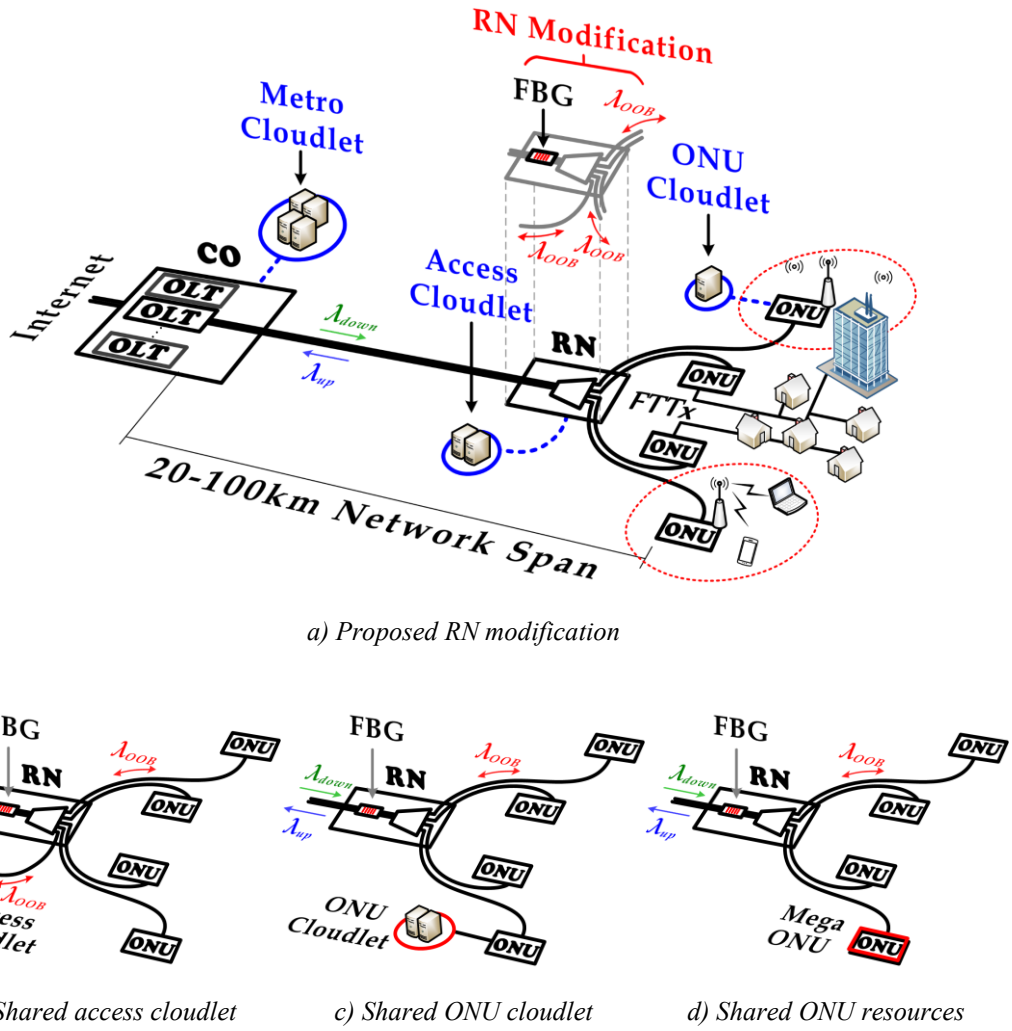


Figure 6.1: Possible cloudlet placements with decentralized RN modification.

Figure 6.1 illustrates the proposed RN modification along with these new cloudlet possibilities. The inter-ONU communications enable the cloudlet to be either a shared access cloudlet, a shared ONU cloudlet, or shared resources of a more advanced ONU.

Figure 6.2 illustrates an example of OOB communications with an access cloudlet using the decentralized framework introduced in Chapter 5. In the figure, ONU_2 and ONU_3 first declare having data that needs offloading by toggling a flag in their outgoing tags and specifying the size of their data similar to the concept explained in Chapter 4. They can then take turns offloading during the next OOB silent period. In the following cycle, the cloudlet itself declares having data

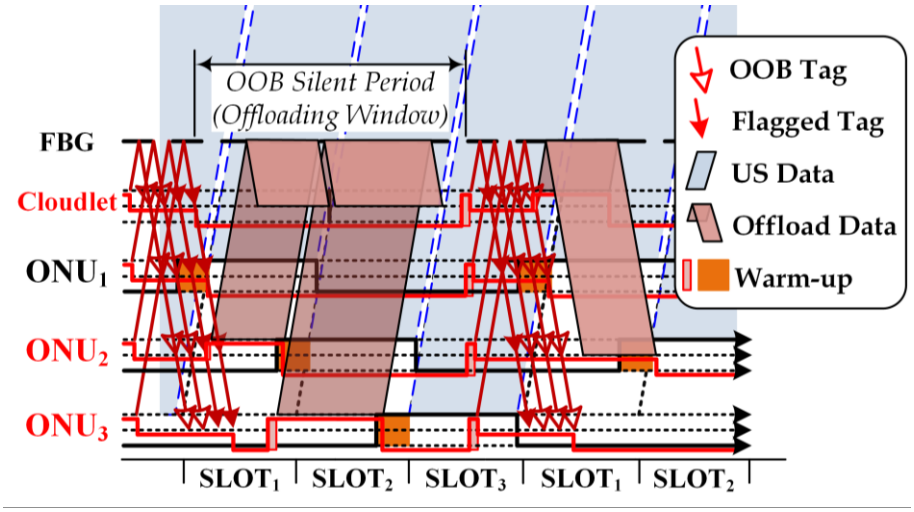


Figure 6.2: Offloading to a shared access cloudlet in SAMAH.

to send, which in this case is a response to ONU_2. The cloudlet therefore behaves the same way that an ONU would in both its signaling and offloading.

6.2 Offloading Negotiations with the Cloudlet

In Chapter 3, we modeled the delays of both centralized and decentralized-based offloading. The delay model assumed that data can be offloaded directly with no negotiations with the cloudlet. In other words, the cloudlet was always ready to accept service requests. A more accurate model must account for cloudlet negotiations, where the device requesting the service would first need to convey three key pieces of information; the task input data size, its computational workload (e.g., CPU cycles) and its delay constraint. The ONU would then relay this request to the cloudlet and must wait for a service acceptance before allowing the device to offload its input data. While negotiations with an access cloudlet are straightforward with inter-ONU communications, negotiations with an indirectly connected access cloudlet using centralized allocation would incur more delays. For instance, if the OLT can negotiate on behalf of an indirectly ONU-cloudlet (i.e., the OLT is constantly updated with the cloudlet's available resources), it would add at least an RTT

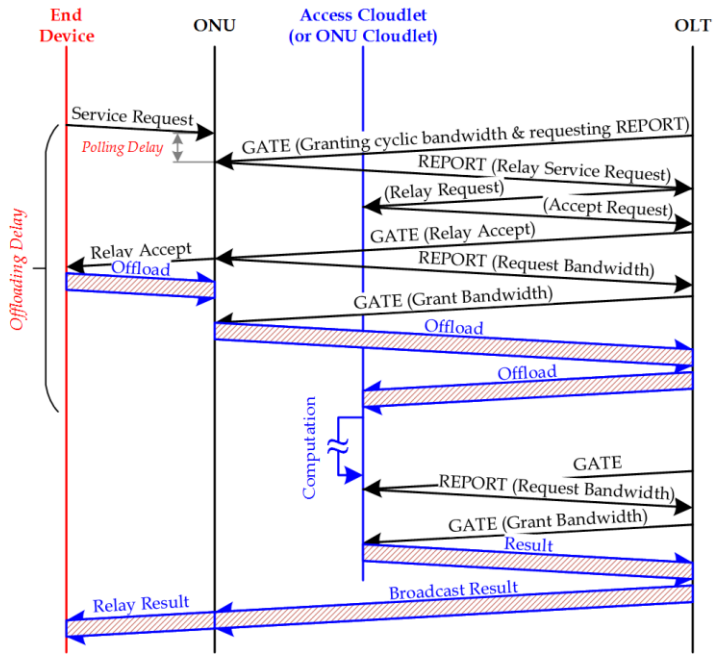


Figure 6.3: Centralized negotiations with indirectly-connected ONU cloudlet.

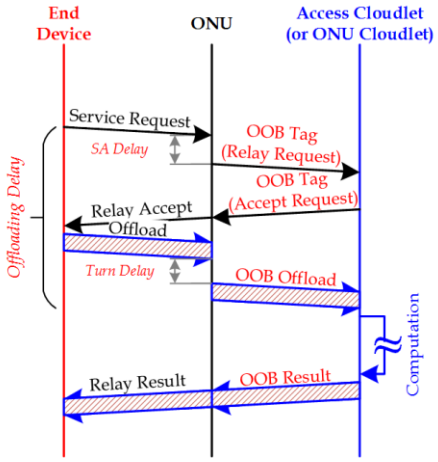


Figure 6.4: Decentralized negotiations with indirectly-connected ONU cloudlet.

to the offloading delay. On the other hand, if the OLT must forward negotiations to the cloudlet, it would add two RTTs as illustrated in Figure 6.3. Still, such propagation delays will not be the dominant offloading bottleneck for cases that have large offloaded data or low transmission rates, as was explained in Chapter 3. Figure 6.4, on the other hand, illustrates direct OOB cloudlet negotiations in the decentralized case, which are much faster than centralized negotiations.

Table 6.2 The Evolution of Decentralized Allocation

	TTACT	TTACT	SAMAH
Chapter	3	4	5
Tagging	default	**UITR**	**DT**
Allocation	online	online	online or offline
Upstream gaps	low chances	minimal	none (best)
Energy efficient	no	yes	yes (best)
Offloading window duration	$W_i - T_{tag}$, $T_{tag} = D_{i,i+1}^{prop} + \frac{L_{tag}}{R_{OOB}}$ (max at $W_i = W_{i,\max}$)	$\sum_{i=1}^{N}\left(W_{i,\max} - T_{tag}\right)/n$, $T_{tag} = D_{i,i+1}^{prop} + \frac{L_{tag}}{R_{OOB}}$ (max at $n=1$)	$\left(\sum_{i=1}^{N} W_{i,\max} - T_{tags}\right)/n$, $T_{tags} = N\frac{L_{tag}}{R_{OOB}} + D_{SC,i}^{prop} - D_{SC,1}^{prop}$ (max at $n=1$)

6.3 The Evolution of Decentralized Allocation

During the course of this dissertation, we have modified and developed decentralized allocation schemes to support fog integration and faster offloading. Table 6.2 summarizes the main features of each decentralized scheme. Starting from TTACT in Chapter 3, we enabled the ONU to offload using the OOB window that is parallel to its upstream window. To maximize the size of this window, the ONU was allowed to reserve its maximum upstream window ($W_i = W_{i,\max}$) even if it was not going to be fully utilized. In Chapter 4, we modified an immediate tagging scheme to force ONUs to exchange media access messages at the beginning of each cycle so that a long silent OOB period may exist afterward. This silent period can then be divided among *n* offloading ONUs, which announce having data to offload by toggling a certain flag in their media access tag frames. Finally, in Chapter 5, we developed a new decentralized framework that is more energy-efficient and offers a slightly longer OOB offloading window. The framework compressed the ONUs' tag exchange period by using a *dense tagging* (DT) scheme that allows interleaved tagging.

Table 6.3 Symbol Definitions and Simulation Values

Symbol	Description	Value
$L_{\max}$	Maximum network span	100km
$L_{i,SC}$	ONU$_i$'s distance from splitter/combiner	0–5km
L_{SC}	Splitter distance from OLT (feeder span)	95km
R_{down}	Downstream transmission rate	10Gbps
R_{up}	Upstream transmission rate	1Gbps
R_{OOB}	Out-of-band channel data rate	1Gbps
R_{user}	Connected users' upstream rate	100Mbps
N	Number of ONUs	16
Q	ONU buffer size	10MB
L	Normalized network load	0.1, 0.5, 0.9
C	Cycle duration	10 – 90ms
T_g	Guard interval	5µs
T_{oh}	Warm-up overhead	2ms, 0.125ms
T_{set}	Doze overhead	760ns, 205ns
P_a	ONU power when fully awake	5.052W, 3.95W
P_d	ONU power during doze mode	3.85W, 3.85W
P_s	ONU power during sleep mode	0.75W

Because the tag exchange was carried out ahead of each cycle, the proposed framework was called *Signaling-Ahead for Media Access Handovers* (SAMAH). SAMAH enabled the bandwidth allocation to be offline, with each ONU having a full view of other ONUs' bandwidth requirements. This, in turn, allowed the realization of various bandwidth allocation schemes and the use of excess distribution. It also helps eliminate any upstream idle gaps when an adaptive cycle is used. The framework was shown to have minimum upstream delays compared to its predecessors.

6.4 Numerical Results

In this section, we first compare between the energy-efficient frameworks developed in Chapter 5 in terms of upstream packet delay, energy conservation, and offloading delays. We then show the effect of applying the more accurate cloudlet negotiation model on the offloading delays. For our simulations, we use the same parameters used in Chapter 5, which are summarized in Table 6.3.

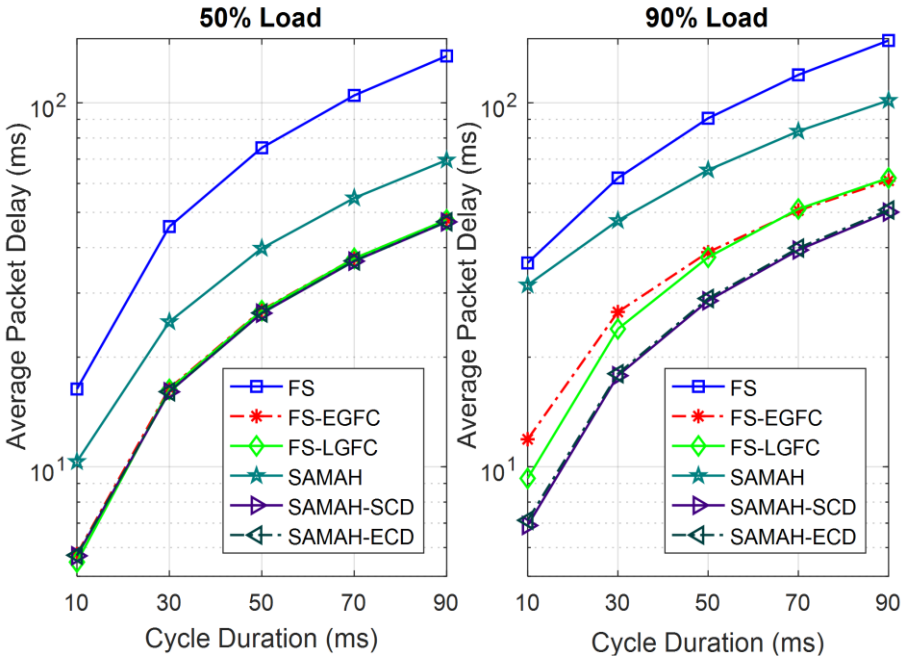


Figure 6.5: Average packet delays at 50% and 90% loads.

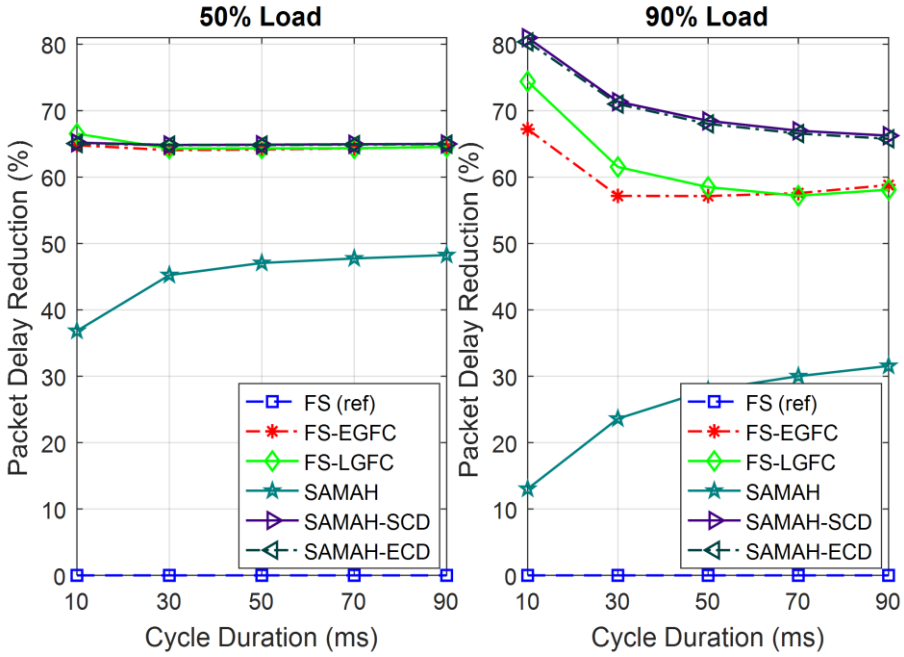


Figure 6.6: Delay reduction at 50% and 90% loads.

6.4.1 Delay Performance

Figures 6.5 and 6.6 show the average upstream packet delays and the achieved delay reductions compared to centralized *fixed-slot* (FS) allocation, respectively, at 50% and 90% loads.

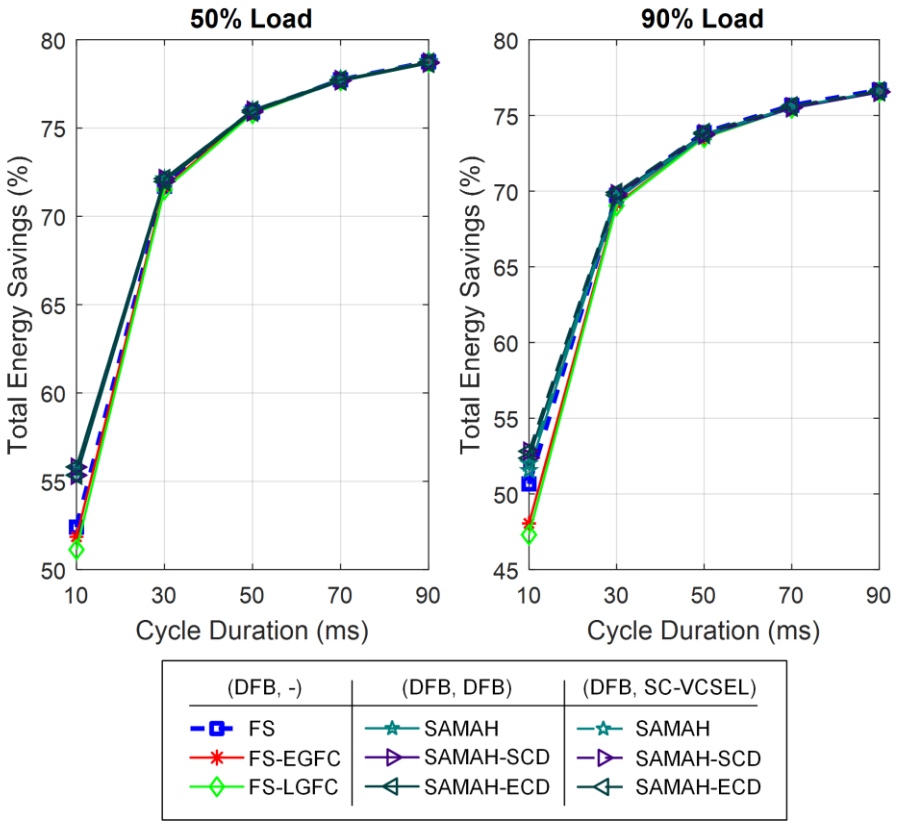


Figure 6.7: Energy Savings vs. cycle duration at 50% and 90% loads.

The two most promising centralized improvements introduced in Chapter 5; *earliest gap-filling and credit* (EGFC) and *largest gap-filling and credit* (LGFC) are also shown in this comparison. Decentralized SAMAH with DT is shown to have significant improvements over centralized FS. Enhancing SAMAH's bandwidth allocation with *sequence and credit distribution* (SCD) or *estimated credit distribution* (ECD) allow it to have similar performance as FS's enhancements under 50% loads (reducing FS's delays by 65%), but make it have 10% of further delay reduction under 90% loads.

6.4.2 Energy Performance

Figure 6.7 shows the energy savings achieved when applying these schemes to a typical PON that has its ONUs always on. The consumptions of both transceivers of decentralized schemes

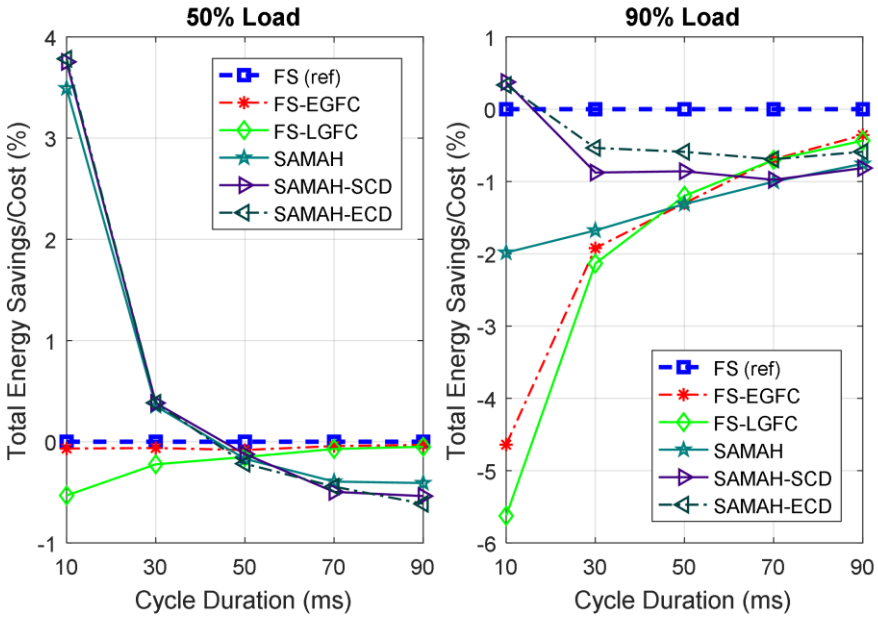


Figure 6.8: Energy Savings vs. cycle duration using a DFB secondary receiver.

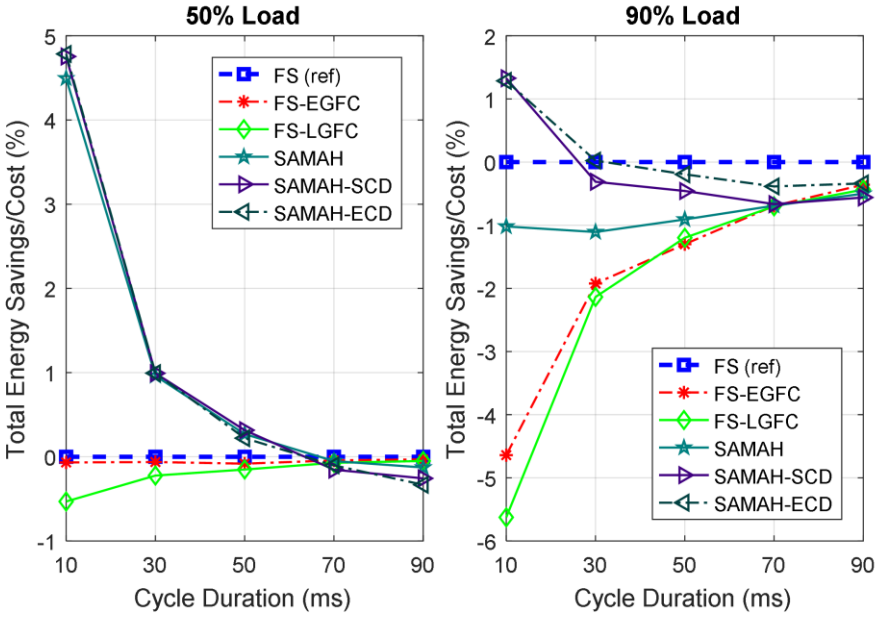


Figure 6.9: Energy Savings vs. cycle duration using a SC-VCSEL secondary receiver.

are considered when the secondary transceiver is either DFB or SC-VCSEL. The energy performance is very similar for all schemes, which achieve higher energy savings at longer cycles and under light network loads. To better visualize the differences in the energy performance of these schemes, we compare their performance against FS. Figures 6.8 and 6.9 show the energy

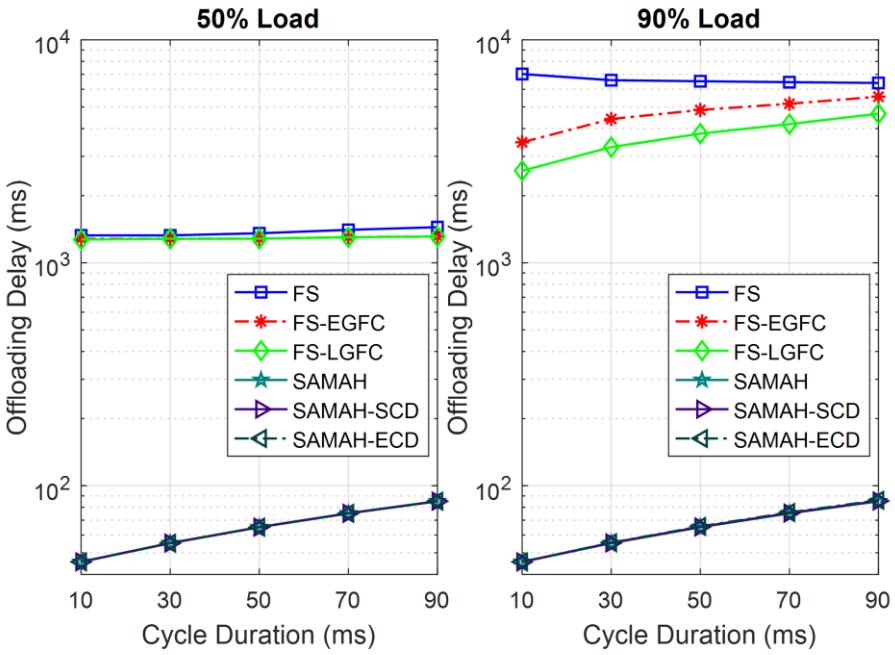


Figure 6.10: Offloading delays at 50% and 90% load.

performance when the secondary decentralized transceiver is either a DFB or an SC-VCSEL laser, respectively. While FS's enhancements always incur more power consumption than pure FS (negative savings), decentralized SAMAH and its enhancements are more energy-efficient, especially under light loads, short cycles, and with SC-VCSEL secondary transceivers.

6.4.3 Offloading Delays

To study the offloading performance of the discussed schemes, Figure 6.10 shows the delays of offloading a 5MB task to a shared access cloudlet. In the decentralized case, the cloudlet is connected with an optical branch similar to ONUs, as discussed in Section 6.1. On the other hand, the cloudlet is assumed to be connected to the RN in the centralized case, employing upstream filtering since this does not require placing additional fiber nor line cards. In centralized allocation, however, an ONU would only be allowed to utilize unused bandwidth to offload data. This is why centralized-based offloading takes much longer than decentralized offloading, which

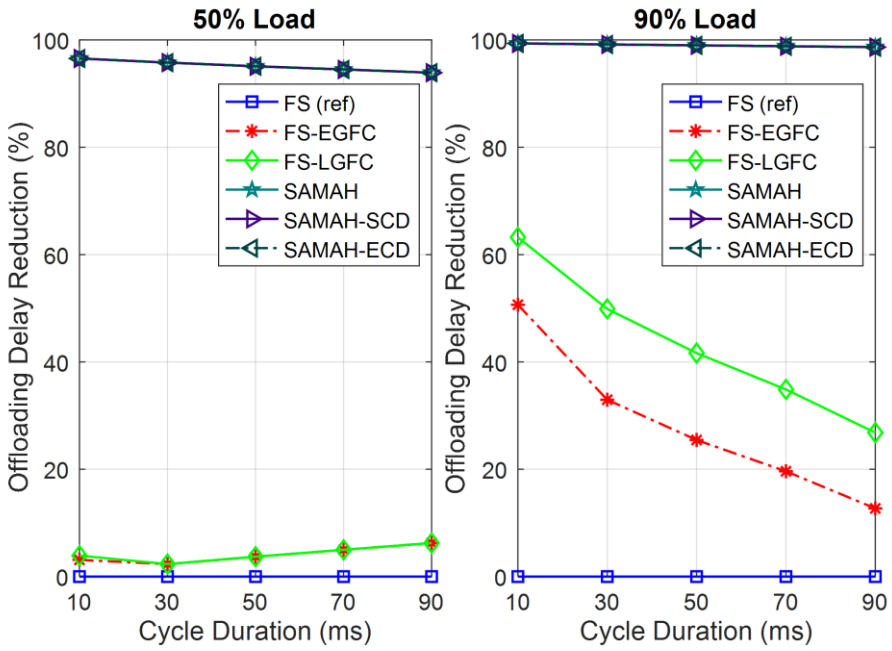


Figure 6.11: Offloading delay reduction compared to FS.

instead utilizes the silent OOB period. Figure 6.11 shows the offloading delay reductions over FS, where FS's enhancements succeed in bringing down its delays by utilizing gaps of lightly-loaded ONUs. Still, decentralized SAMAH has much lower delays for employing direct OOB offloading, making its offloading delays less than 1% of centralized FS under heavy loads. Note however that, in the case of concurrent offloading, decentralized offloading delays will increase accordingly since the silent period will be divided among the offloading ONUs. On the other hand, centralized offloading will be less affected by concurrent offloading since there is no bandwidth contention except for FS's enhancements, which would have less upstream gaps to utilize.

6.4.4 Size of Offloaded Data

Figure 6.12 illustrates the effect of changing the size of offloaded data on the offloading delays using a 50ms cycle. Because centralized offloading is bound by the unused excess in the ONUs' slots, it needs more cycles for offloading, especially under heavy loads. On the other hand,

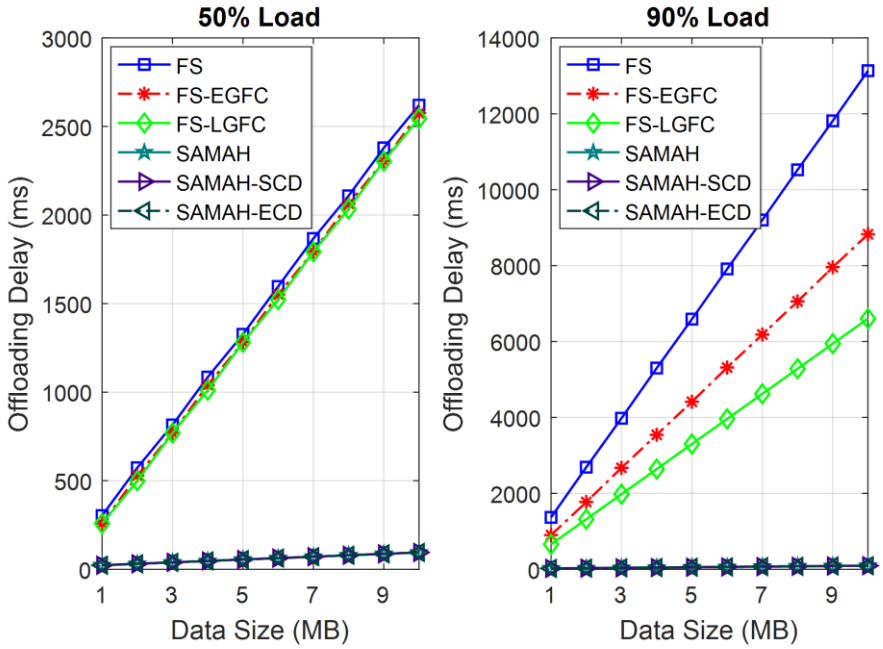


Figure 6.12: Effect of offloaded data size at 50ms.

decentralized offloading offers much longer offloading windows regardless of the network load by utilizing the OOB silent duration, making it much better suited for offloading huge tasks and data.

6.4.5 The Performance Cost of Energy-Efficiency

In this section, we compare the performances of the aforementioned energy-efficient frameworks against their adaptive-cycle counterparts. This enables us to see the tradeoffs in the network performance when employing energy conservation through fixed long cycles. We compare the schemes in terms of upstream delays, throughput, energy savings, and offloading. For centralized allocation, we refer to fixed-slot allocation as *interleaved polling with fixed-slot* (IP-FS) and compare it with *interleaved polling with an adaptive cycle* (IP-AC). We also examine it with both EGFC and LGFC enhancements. For decentralized allocation, we refer to the SAMAH framework as *Signaling-Ahead with a fixed cycle* (SA-FC) and compare it with *Signaling-Ahead with an adaptive cycle* (SA-AC). We also study it with its SCD and ECD enhancements.

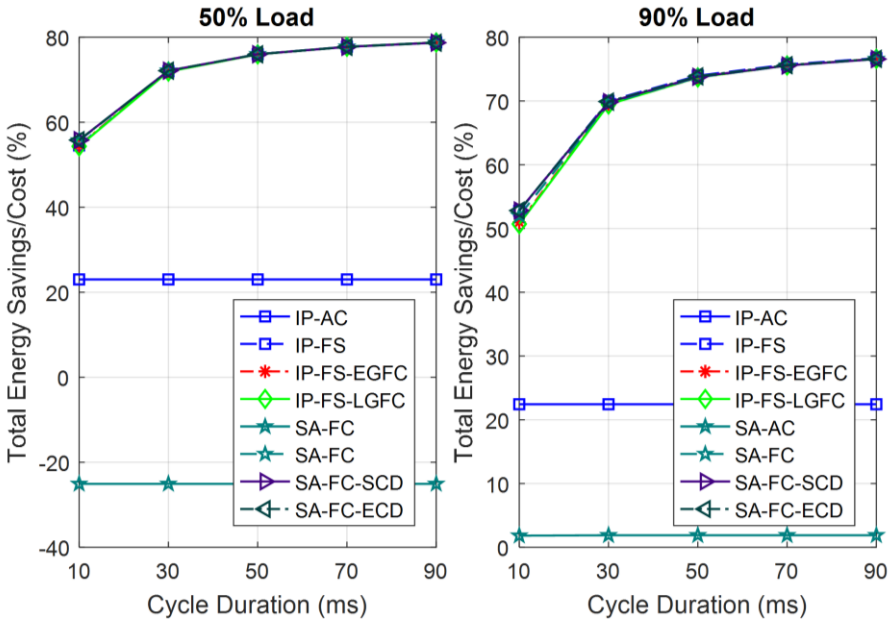


Figure 6.13: Energy performance.

The energy performance of the discussed schemes is illustrated in Figure 6.13, where it can be observed how adaptive cycles do not give ONUs enough time to sleep and therefore result in much less energy savings. However, by enabling the ONUs to switch to doze mode whenever possible, centralized IP-AC still achieves some energy savings (~22-23%) but they are much less than its fixed cycle counterpart. On the other hand, decentralized SA-AC is shown to have more than 20% of additional energy consumption under light loads (negative energy savings) due to employing additional transceivers and using cycles that are often too short to allow it to switch any of it transceivers to sleep mode. Under heavy loads, however, the cycle may often extend allowing for some energy savings (~2%). Note that centralized cycles are lower-bounded by the RTT, which allows them to achieve better savings under an adaptive cycle configuration. Still, it is clear that adaptive cycles cannot achieve much energy savings, especially for decentralized allocation, which employs additional transceivers and usually has effective cycles that are too short to allow for noticeable energy savings.

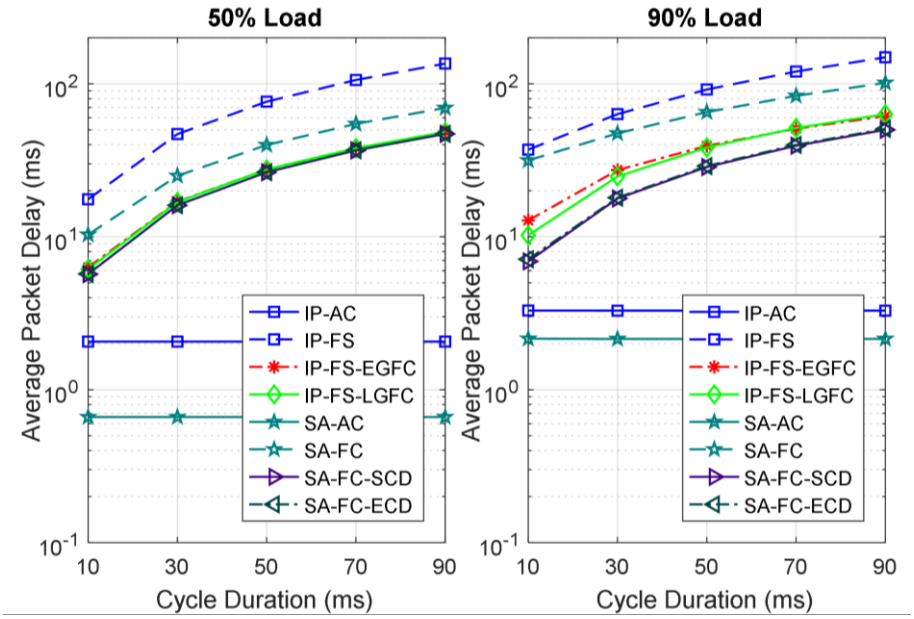


Figure 6.14: Upstream packet delays.

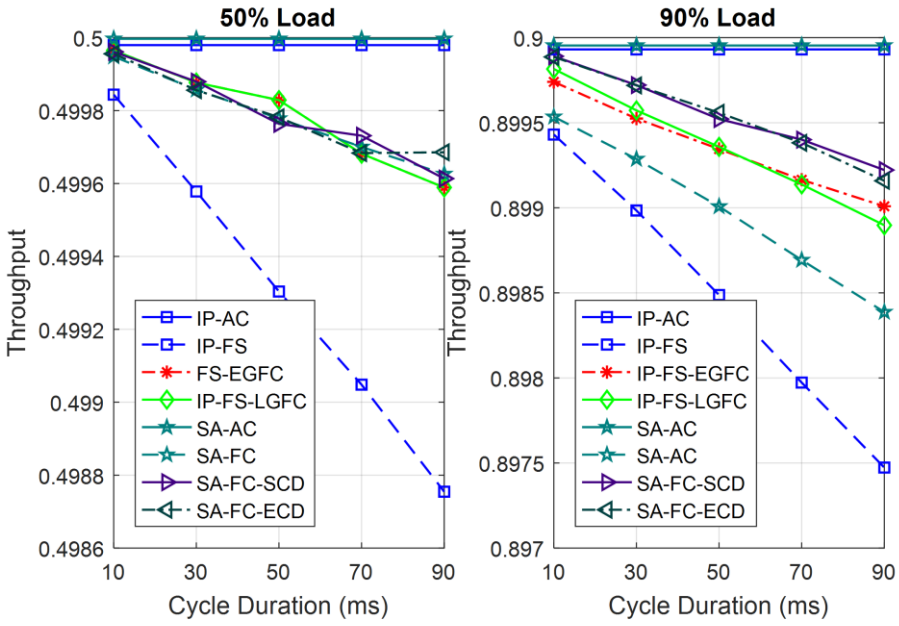


Figure 6.15: Throughput performance.

We next examine the network performance in terms of both upstream packet delays and throughput, which are illustrated in Figures 6.14 and 6.15, respectively. While using fixed long cycles enables considerable savings, they also lead to degrading the network performance. IP-FS

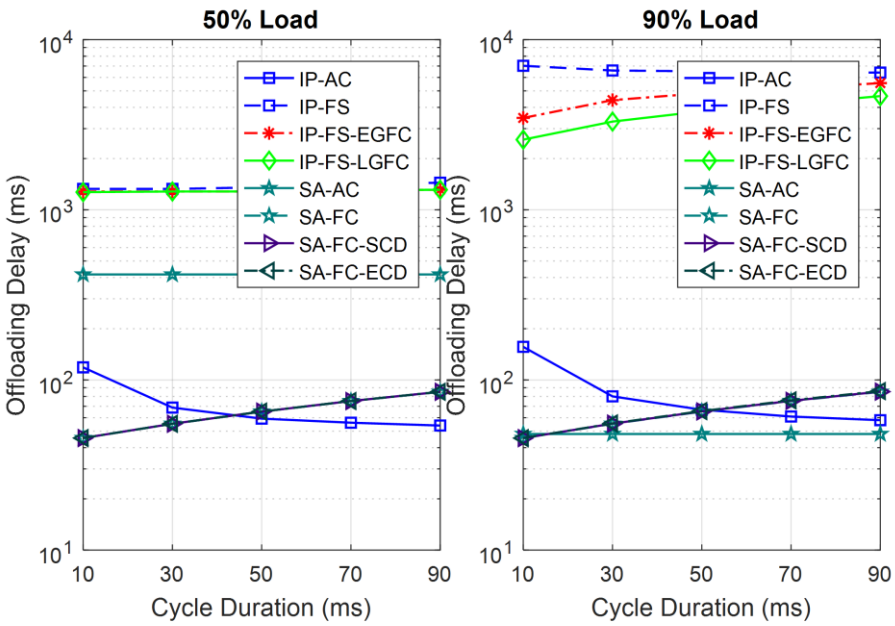
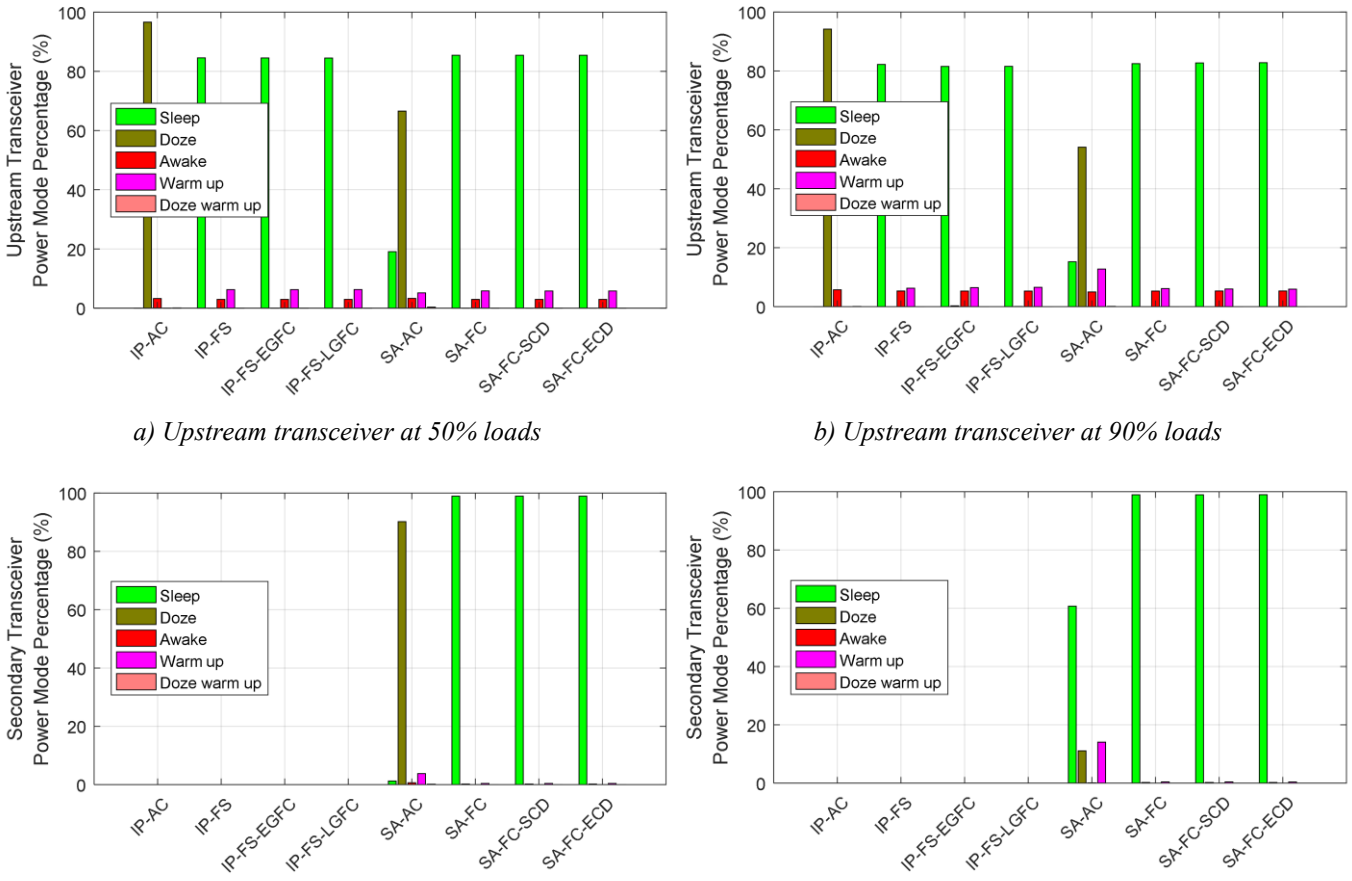


Figure 6.16: Offloading delays for a 5MB task.

has the worst performance, whereas SAMAH's enhancements are the closest thing to the high performance of the adaptive-cycle schemes, especially under heavy loads.

To examine the tradeoffs in the offloading performance, Figure 6.16 shows the offloading delays of a 5MB task using an RN access cloudlet. Because offloading is carried out in a different manner in each paradigm, its performance is affected differently according to the network load or the chosen cycle configuration. In centralized allocation, offloading is done through the ONU's excess bandwidth, which can take many cycles depending on the excess available and the size of the offloaded data. Therefore, both light loads and long cycles lead to a less number of offloading cycles for both adaptive and fixed cycles since more excess is available. Using an adaptive cycle results in even faster offloads since the ONU does not have to wait as long before resuming its offloaded data. Moreover, when using fixed cycles for energy conserving, FS enhancements result in faster offloads than FS since an ONU may be granted additional bandwidth belonging to underutilized ONUs. This is also very beneficial under heavy loads, where there may not be much



a) Upstream transceiver at 50% loads

b) Upstream transceiver at 90% loads

c) Secondary transceiver at 50% loads

d) Secondary transceiver at 90% loads

Figure 6.17: Percentages of power modes for ONU transceivers at a 30ms cycle.

excess available in an ONU's slot for offloading. On the other hand, offloaded data is transmitted out-of-band in the decentralized case. Offloading with fixed cycles is therefore not affected by the network load but slightly affected by the cycle duration since an ONU has to wait longer before accessing the OOB channel. Because adaptive cycles shrink under light loads, they do not allow for a large enough OOB window, which is why offloading is much faster under heavy loads for decentralized allocation with adaptive cycles. SAMAH's enhancements do not affect the offloading performance since they are merely enhancements for bandwidth allocation on the upstream channel. Note however how decentralized fixed cycles have comparable offloading

performances with centralized adaptive cycles. Of course, the former also have much better energy savings at the cost of higher upstream packet delays.

Because a 30ms cycle here appears to be a reasonable compromise between network performance and energy savings, Figure 6.17 shows the duration percentage of each power mode of the ONUs' transceivers at a 30ms cycle under 50% and 90% network loads, respectively. The figure provides insights on how using fixed cycles lead to long sleep durations even for additional transceivers in decentralized allocation, in which adaptive cycles enable it to switch to sleep mode under heavy loads.

6.5 Conclusions

In this chapter, we compared the proposed decentralized energy-conserving framework against its centralized counterpart. Numerical results have shown that, even with employing additional ONU transceivers, the proposed decentralized framework has the overall best performance in terms of both packet delays and energy efficiency. Moreover, it also supports exclusive cloudlet architectures that enable fog and edge computing paradigms to be realized in such access networks with much lower offloading delays. Comparing the energy-conserving frameworks against their adaptive cycle counterparts demonstrated the tradeoffs associated with energy efficiency. Still, the energy-efficient decentralized framework was shown to have the closest network performance to that of adaptive schemes.

Chapter 7

Conclusions and Future Research

7.1 Concluding Remarks

This dissertation presented a study of some of the considerations and challenges that face integrating fog and edge computing with optical access networks. The integration was studied from both bandwidth allocation and energy efficiency perspectives in order to identify the potential bottlenecks and propose possible solutions. As such, our work addressed many challenges facing access networks that are usually studied separately in the literature.

While fog computing was realized through placing a cloudlet in the network, edge computing was proposed to be done through the resources of edge devices and the network itself. The offloading performance was then studied when the underlying allocation is either centralized or decentralized. A theoretical analysis of the offloading performance was developed and compared against simulation results using a developed MATLAB system-level simulator. The performance of the proposed schemes was then examined by observing a number of metrics, such as the upstream and offloading delays in addition to the effect of offloading on upstream traffic. The MATLAB simulation was later revised to incorporate different power modes and measure several energy related metrics.

From an underlying bandwidth allocation perspective, we examined how offloading can be supported in each allocation paradigm and investigated the various parameters that may affect its

performance. To maintain the low complexity and low cost of the ONUs in centralized allocation, cloudlet offloading was done through excess upstream bandwidth. On the other hand, the additional inter-ONU communications channel in decentralized allocation was modified to support out-of-band offloading. This was shown to achieve significant gains in the offloading performance in addition to the improved network performance due to decentralized allocation. Not only did decentralized-based offloading become independent of upstream traffic but also the offloading performance was not degraded as much as centralized-based offloading when the size of the offloaded data was increased.

To address the requirement of energy-efficiency, the ONUs needed to be able to switch to sleep mode in order to reduce their power consumption. This required modifying bandwidth allocation to have fixed long upstream cycles so that the ONUs have viable sleeping durations as well as enough time for warming up. This cycle elongation, however, led to severe degradation in network performance, which is why part of the work was focused on modifying and proposing new schemes that improve the network performance while maximizing energy savings.

Because the performance of centralized allocation was already degraded in long-reach PONs, it faced a great challenge to be able to perform well under energy-conserving frameworks. Moreover, not all centralized allocation schemes were suitable for conserving energy in LR-PONs. For instance, multithread polling was designed to withstand the degradation of centralized allocation in long-reach networks by creating multiple interleaved parallel threads of communications between the ONUs and the OLT. Yet, these frequent threads of communications prevent the ONUs from switching to sleep mode unless excessively long cycles are used to account for the multiple sleep durations for each ONU and their corresponding 2ms wake-up overheads. Multithread polling therefore cannot be used for energy-conservation in PONs. Instead, we proposed using credit allocation and gap-filling on top of fixed slot allocation to improve its

performance. With the former, ONUs are allocated extra bandwidth in addition to their requests, which allows newly arriving packets (not yet reported) to be accommodated in their next slots. Gap-filling, on the other hand, enables allocating unutilized bandwidth from lightly-loaded ONUs as extra windows to heavily-loaded ones. Gap-filling may therefore be considered as a temporary extra thread of communication for a heavily-loaded ONU that is created whenever needed and may therefore be regarded as an energy-conserving version of multithread polling.

On the other hand, decentralized allocation proved to have better network performance as well as lower offloading delays under energy-conserving frameworks. It also allowed for novel cloudlet architectures that do not require further fiber deployments nor active switching in the remote node. Because the improved network and offloading performance of decentralized allocation came at the cost of placing additional ONU transceivers and utilizing an additional channel, its energy-efficiency faced a unique challenge. Still, we were able to develop a novel framework that enabled decentralized allocation to have very comparable energy efficiency with centralized allocation.

Based on the findings of this study, we arrived at certain conclusions as follows. Although offloading in-band has some advantages, such as not requiring additional transceivers and supporting a transparent offloading approach, it was found to constitute a bottleneck on the offloading performance due to its contention with upstream traffic, especially under heavy network loads. It also affects the delay performance of upstream traffic by the amount of cycle elongation. In-band offloading also requires offloaded traffic to be filtered from regular traffic on the upstream and additionally requires injecting cloudlet responses with OLT transmissions on the downstream. For these reasons, we believe that offloading will eventually have to be done out-of-band on a separate channel to ensure better and more reliable offloading performance with no side effects on upstream packet delays. It also enables the use of pre-existing fiber deployments and does not

Table 7.1 Cost vs. Performance for Both Allocation Paradigms.

		Centralized	Decentralized
Cost	Bandwidth allocation	**None**	FBG + additional ONU transceivers
	Cloudlet communications	Additional ONU transceivers + fiber or active switching	**None**
	Energy (running cost)	**Slightly worse** (additional energy is needed to improve network performance)	**Slightly better**
Performance	Network performance	**Worse** (affected by network span)	**Better**
	Offloading performance	**Worse**	**Better** (independent of upstream traffic)
	Offloading bottlenecks	Heavy network loads and size of offloaded data (due to contention with upstream traffic)	Concurrent offloading
	ONU-cloudlet sharing	**Indirect** (through OLT)	**Direct**

require adding filtering and switching devices to the network. In fact, some recent studies proposed placing three transceivers within ONUs to enable them to communicate with field cloudlets and shared access cloudlets in addition to their communications with the OLT [90]. However, with no OLT access to these channels, ONU cloudlet transmissions will then have to be coordinated by the cloudlets themselves, making their resource allocation neither centralized (not done by the OLT) nor decentralized (not done by the ONUs). On the other hand, the proposed decentralized framework enables the ONUs to manage both upstream and offloading traffic. The inter-ONU communications channel also allows for unique cloudlet placements, where field cloudlets may only be connected to some ONUs but still be accessible by other ONUs. Table 7.1 highlights the performance aspects of each allocation paradigm along with its corresponding costs that may either be required for enabling cloudlet communications or bandwidth allocation.

During the course of our work, we came to see how decentralized-based bandwidth allocation is very analogous to fog and edge computing. While the former proposes that edge devices manage their upstream media access and cloudlet offloading, fog and edge computing promote edge devices to do more processing at the edge of the network. Both concepts therefore assign more roles to edge devices in order to reduce delays and network congestions. As the fog computing concept grows necessary to complement cloud computing, so may decentralized-based resource allocation. We therefore believe that decentralized allocation and its edge-to-edge communication go hand-in-hand with these computing concepts since they truly enable computing to be carried out at the edge in a timely fashion. Examining the potential of decentralized allocation in the integration of fog and edge computing was therefore an important contribution of this work, which may also serve as the basis for future research topics, as discussed in the next section.

7.2 Future Research

During the course of our research, several interesting research directions were risen that can expand upon this work. We discuss some of them in the following:

- *Smart cycles and deep sleep:* Throughout our simulations, upstream network traffic was modeled as self-similar traffic, similar to most studies in the field. The ONUs were then considered to be identical yet independent traffic sources. In other words, the network traffic load at any given point was the sum of the traffic generated by N identical yet independent traffic streams. In reality, however, not all ONUs generate the same amount of traffic nor are equally active. Some ONUs may, in fact, stay idle for long periods of time. In such cases, it will be inefficient to treat the ONUs equally under the employed energy conservation framework. Instead, inactive ONUs may be taken out of the transmission sequence for a number of consecutive cycles, where they can switch to a deep sleep state.

The upstream cycle would then become much shorter since it will only contain active ONUs. This will lead to a significant improvement in the delay performance. To study this improvement, new traffic models are needed to reflect such cases.

- *Hybrid service composition:* In Chapter 4, we compared between centralized and decentralized-based service composition. Even though the latter was shown to have much better performance, the former allows access to services and resources in other access zones and may therefore also have lower service rejection ratios. To capture the benefits of each allocation paradigm, a hybrid service composition framework needs to be developed, which would allow some decentralized-based services to be carried out in a timely fashion while allowing others to be forwarded to other parts of the network. A management framework needs to be developed for such a scheme, which must be both scalable and reliable.
- *In-band offloading and SDN:* Although out-of-band offloading was shown to have better performance with no contention with upstream traffic, in-band offloading may still be useful in cases where access networks are underutilized. With abundant excess bandwidth on the upstream, there may be no need to use additional channels and place extra transceivers for supporting cloudlet communications. Still, this will require differentiation between normal upstream traffic and offloaded traffic based on the users' service level agreements and underlying delay constraints for each type of traffic. This kind of differentiation may be carried out using *software-defined networking* (SDN), which may then require redesigning the network and determining where the controller will be placed. Moreover, depending on the nature of the applications in the access network (e.g., whether the access network serves a smart grid or wireless mobile devices), the delay constraints for offloaded traffic may vary, which may present different challenges.

Appendix A

On Modeling and Simulating PONs

This appendix sheds some light on the work done to model and simulate passive optical networks, on the data-link layer, using MATLAB.

A1. Introduction

The system under study in this work is a multipoint-to-point network, connecting multiple ONUs to a single OLT. Each ONU employs a finite data buffer, which receives aggregated network traffic from the user side. Such a network can be simulated as a discrete-event system, in which the state variables change at a discrete set of points in time, allowing the simulation to jump through time periods that have no event occurrences.

A2. Defining System Events

The entity, that is the object of interest in our system, is the Ethernet frame, which keeps advancing from one network device to another. Since we focus on upstream media access, a frame could either be an upstream data frame or a media access control frame. A data frame has two main events; its arrival at an ONU and its departure from its buffer when it is transmitted to the OLT. The frame arrival event results in the frame either being queued in the buffer or being dropped if

the buffer is full at the time of its arrival. On the other hand, the frame departure can only take place at the time the ONU is granted media access after all preceding frames in the buffer have been transmitted. It then follows that the main event, which changes the system state (i.e., upstream channel status), is the ONU being granted media access. This event takes place according to the interpretation of the control frames. Therefore, instead of making the simulation jump in time by data frame arrival events, it can jump to events where an ONU gains access to the upstream channel. At these times, the simulation can retrospectively account for any minor events that have already taken place (i.e., the arrival of data frames at the ONUs).

A3. Defining Performance Metrics

The system performance metrics in our study are all related to the entity of interest, which is the Ethernet frame. The first measure is the packet pre-transmission delay, which is the time a data frame has to wait before being transmitted. This delay can further be broken down into its elemental delays such as the reporting delay (the time the frame waits before being reported), a grant delay (the time it has to wait since being reported until it is granted transmission), and a queuing delay (the time it waits to be transmitted after being granted transmission). To compare between the different bandwidth allocation schemes in terms of packet delays, a high target number of packets (e.g., 20 million) must be set before the average delay is computed for each scheme. This is because fixing a simulation time for all schemes may not be as accurate since the simulation may yield a different number of transmitted packets for each scheme.

The second performance metric is the network throughput, which is the amount of data transmitted over a certain period of time. It follows from its definition that, unlike packet delays, a fixed simulation duration must be set in order to compare between the different schemes in terms

of their throughput. Note that the throughput is different from the upstream utilization, which reflects how efficient the upstream channel is used.

The final performance measure is the energy consumption. This not only depends on the status of the ONU's transceiver, but it also depends on the bandwidth allocation scheme itself. This is because some schemes do not support sleep-mode and can only allow for a dozing period, where the ONU can turn off its transmitter. Other schemes allow the ONU to know the next time of its media access early on. The energy consumption therefore requires monitoring both the physical status of the ONU's transceiver (whether it is transmitting or not) and the logical state of the ONU (whether it is waiting for a control message or not). Depending on the transceiver's power mode, the power consumption is recorded over its duration. Moreover, similar to the throughput computation, a fixed simulation duration must be set in order to compare between the allocation schemes in terms of their energy consumption.

A4. Simulation Layout

Figure A.1 illustrates the general layout of our simulator. As shown in the figure, the first step in the simulation is to define our parameters, such as the distances of the ONUs from the OLT, the network transmission rates, the network traffic loads, and the bandwidth allocation parameters. After these parameters have been set, the simulation is carried out through three overlapping loops; a loop for each simulated network load, a loop for each cycle duration, and an inner loop for simulating each bandwidth allocation scheme under a certain load and a specified maximum upstream cycle duration.

The outer loop can be called *the traffic-loop* and is mainly responsible for making sure that the access traffic trace-files exist for each simulated network load. If no previously generated trace-files are found for a certain load, a traffic generator module is responsible for generating N streams

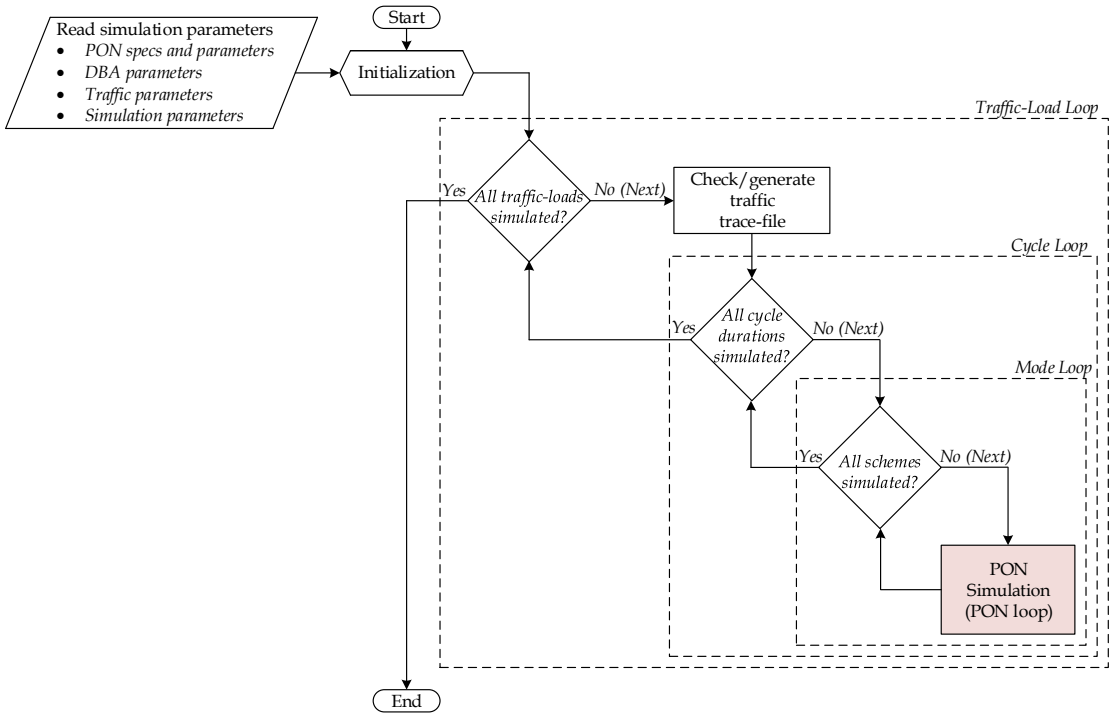


Figure A.1: Simulation layout for comparing different allocation schemes.

of self-similar traffic for the *N* ONUs such that the sum of these streams forms the desired network load. The streams are random and independent and can either be symmetric or asymmetric streams. Each stream is represented in a trace-file by two arrays; one for the packets lengths and another showing their arrival times. More arrays can be added, if needed, to reflect other frame attributes such as its class or its delay constraints. Additionally, the lengths of the frames can be picked from a uniform distribution or can follow more realistic models, where certain frame sizes are more likely to occur. Such realistic distributions are widely known as the *Internet Mix* (IMIX) [91]. Working with trace-files enables them to be reused and allows simulating different allocation schemes with the exact same traffic streams.

The actual PON simulation module lies within the three aforementioned loops in order to be performed for each allocation scheme, at each cycle duration, at each target load. Figure A.2 illustrates the layout of the PON simulation module. After its initialization, the PON module is

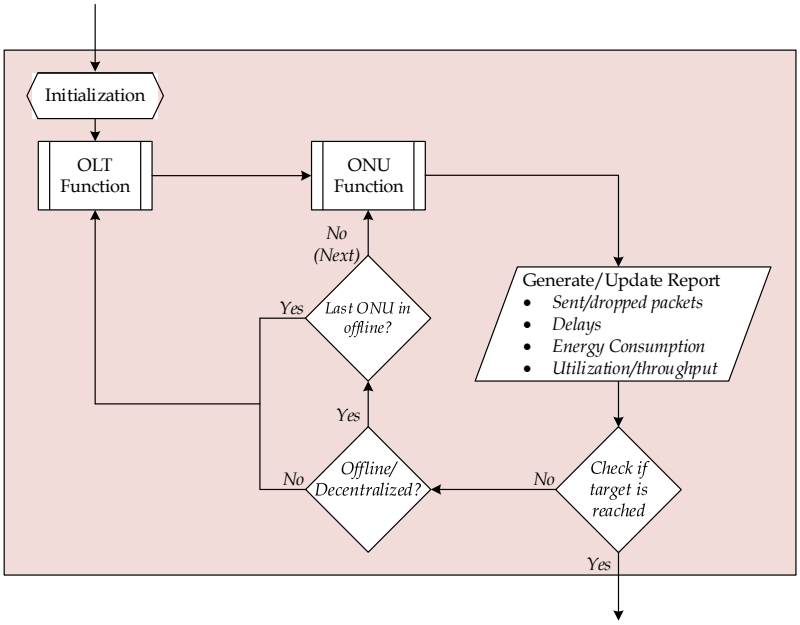


Figure A.2: Layout of the PON module.

basically a while loop that rotates between two subroutines resembling the OLT and any given ONU. The loop terminates only when the simulation target is reached, whether it is a simulation duration or a number of transmitted packets.

The OLT function is different for each allocation paradigm. In centralized allocation, it sets a transmissions window for each ONU in each cycle either based on its received report (online scheduling) or all cyclic reports (offline scheduling), depending on the simulated scheme. The PON simulation starts by making the OLT send the ONUs zero-sized grants in order to initiate their reports. As the OLT starts granting the ONUs media access, it keeps track of the next upstream channel availability. In a decentralized allocation, on the other hand, the OLT function is relatively simpler. Its main function would be to initialize the ONUs by setting their transmission sequence and maximum transmission windows. After the OLT sends the ONUs these control parameters, it will no longer be involved in the upstream media access. Instead, the PON loop would keep rotating between the *N* ONUs by calling the ONU function for each ONU.

Similar to the OLT function, the ONU function greatly differs for each allocation paradigm. In centralized allocation, the ONU function is triggered each time an ONU receives a grant. Because the simulation is designed to advance to grant events rather than packets arrival events, the first step of the ONU function is to retrospectively check for all packets that had arrived since its last grant event dropping those that had arrived after the buffer is full. The ONU function will then check to see how many of its previously buffered packets fit within its granted transmission window and compute their grant and queuing delays as they are transmitted one after another. Finally, the ONU function will check the buffer occupancy up until this moment, since new packets may have arrived while the payload is being transmitted, before sending its new report to the OLT and starting a report delay counter for the newly reported packets. In decentralized allocation, after the ONUs receive their initialization parameters, the ONU function is triggered each time an ONU receives a tag frame. The function is then split into two subroutines; media access and tagging. This is because these two functionalities are decoupled in immediate tagging and dense tagging schemes, in which an ONU may only have to send a tag without accessing the media.

A5. Simulating Offloading

In order to simulate offloading and study its effects on upstream traffic, another simulation layer is required. After an initial simulation for all schemes is performed, random points in time are chosen from it to inject offloaded data in another simulation run. This second simulation uses some of the statistics reordered from the first, such as:

- the ONUs' excess bandwidth in their allocated slots (at the selected points in time and for as many consecutive cycles as needed),
- the effective cycle durations,
- the original pre-transmission packet delays (measured at the selected points in time).

That way, the second simulation would be able to provide a clear contrast between the performance with and without offloading. Moreover, offloaded data is always assumed to be fixed in size, but encapsulated in subsequent Ethernet frames. The data is then sent either in-band (centralized case), after requesting additional bandwidth from the OLT, or sent out-of-band (decentralized case) by making use of the silent period on the additional channel. While the data is being offloaded, the higher-layer simulation records the new cycle durations and pre-transmission delays along with the offloading delays themselves (which are measured separately from regular upstream traffic pre-transmission delays). This process is then repeated as many times as necessary to capture accurate estimates of the offloading delay and its various effects under different network loads.

References

[1] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, 2016.

[2] C. C. Byers, "Architectural Imperatives for Fog Computing: Use Cases, Requirements, and Architectural Techniques for Fog-Enabled IoT Networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 14–20, 2017.

[3] S. Yang, "IoT Stream Processing and Analytics in the Fog," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 21–27, 2017.

[4] S. Petridou, S. Basagiannis, and L. Mamatas, "Formal Methods for Energy-Efficient EPONs," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 246–259, 2018.

[5] Cisco, "The Zettabyte Era: Trends and Analysis," *White Paper*, 2017.

[6] Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017-2022," *White Paper*, 2019.

[7] K. Bourg, S. Ten, R. Whitman, J. Jensen, and V. Diaz, "The Evolution of Outside Plant Architectures Driven by Network Convergence and New PON Technologies," in *Proc. Optical Fiber Communication Conference (OFC)*, 2017.

[8] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "MCloud: A Context-Aware Offloading Framework for Heterogeneous Mobile Cloud," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 797–810, 2017.

[9] G. Kramer, "MPCP+: A Proposal for Channel Bonding at MAC Control Sublayer," *IEEE P802.3ca Task Force Meeting*, Macau, China, 2016.

[10] D. Van Veen and V. Houtsma, "Bi-directional 25G-50G TDM-PON with Extended Power Budget using 25G APD and Coherent Amplification," in *Proc. Optical Fiber Communication Conference (OFC)*, 2017.

[11] S. B. Hussain, W. Hu, H. Xin, A. M. Mikaeil, and A. Sultan, "Flexible Wavelength and Dynamic Bandwidth Allocation for NG-EPONs," *IEEE/OSA J. Optical Communications and Networking*, vol. 10, no. 6, pp. 643–652, 2018.

[12] O. Strobel, "Optical and Microwave Technologies for Telecommunication Networks," *Wiley*, 2016.

[13] F. Carvalho and A. Cartaxo, "Multisignal OFDM-based Hybrid Optical-Wireless WDM LR-PON with Colorless ONU," *IEEE Photonics Technology Letters*, vol. 27, no. 11, pp. 1193–1196, 2015.

[14] M. Cen, J. Chen, V. Moeyaert, P. Megret, and M. Wuilpart, "Advanced Transmission-Reflection-Analysis (TRA) System for Long-Reach Passive Optical Network Monitoring," in *Proc. 17th International Conference on Transparent Optical Networks (ICTON)*, 2015.

[15] S. Zhang, W. Ji, X. Li, K. Huang, and Z. Yan, "Efficient and Reliable Protection Mechanism in Long-Reach PON," *IEEE/OSA J. Optical Communications and Networking*, vol. 8, no. 1, pp. 23–32, 2016.

[16] Y. Chen et al., "MC-CDMA Enhanced LR-PON using Widely Wavelength Lockable FPLD with Low Facet Reflectance," *IEEE/OSA J. Optical Communications and Networking*, vol. 9, no. 9, pp. 747–755, 2017.

[17] D. P. Shea and J. E. Mitchell, "A 10-Gb/s 1024-way-split 100-km Long-Reach Optical-Access Network," *J. Lightwave Technology*, vol. 25, no. 3, pp. 685–693, 2007.

[18] H. Song, B. W. Kim, and B. Mukherjee, "Multi-thread Polling: A dynamic bandwidth distribution scheme in long-reach PON," *IEEE J. Selected Areas in Communications*, vol. 27, no. 2, pp. 134–142, 2009.

[19] O. Cappe, E. Moulines, J.-C. Pesquet, A. P. Petropulu, and X. Yang, "Long-range Dependence and Heavy-tail Modeling for Teletraffic Data," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 14–27, 2002.

[20] T. Holmberg, "Analysis of EPONs under the Static Priority Scheduling Scheme with Fixed Transmission Times," in *Proc. 2nd Conference on Next Generation Internet Design and Engineering (NGI)*, 2006.

[21] J. Zheng and H. T. Mouftah, "A Survey of Dynamic Bandwidth Allocation Algorithms for Ethernet Passive Optical Networks," *J. Optical Switching and Networking*, vol. 6, no. 3, pp. 151–162, 2009.

[22] H. Byun, J.-M. Nho, and J. Lim, "Dynamic Bandwidth Allocation Algorithm in Ethernet Passive Optical Networks," *Electronic Letters*, vol. 39, no. 13, pp. 1001–1002, 2003.

[23] Y. Zhu and M. Ma, "IPACT with Grant Estimation (IPACT-GE) Scheme for Ethernet Passive Optical Networks," *J. Lightwave Technology*, vol. 26, no. 14, pp. 2055–2063, 2008.

[24] G. Kramer and B. Mukherjee, "IPACT a dynamic protocol for an Ethernet PON (EPON)," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 74–80, 2002.

[25] S. Choi, "A new multi-thread polling based dynamic bandwidth allocation in Long-Reach PON," in *Proc. 12th International Conference on Optical Internet (COIN)*, 2014.

[26] A. Mercian, M. P. McGarry, and M. Reisslein, "Offline and Online Multi-thread Polling in Long-Reach PONs: A critical evaluation," *J. Lightwave Technology*, vol. 31, no. 12, pp. 2018–2028, 2013.

[27] R. Bushra, M. Hossen, and M. M. Rahman, "Online Multi-thread Polling Algorithm with Predicted Window Size for DBA in Long Reach PON," in *Proc. 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2016.

[28] J. Chen, M. De Andrade, B. Skubic, J. Ahmed, and L. Wosinska, "Enhancing IPACT with Limited Service for Multi-thread DBA in Long-Reach EPON," in *Proc. Asia Communications and Photonics Conference and Exhibition (ACP)*, 2010.

[29] A. Helmy and H. Fathallah, "Taking Turns with Adaptive Cycle Time: a Decentralized Media Access Scheme for LR-PON," *J. Lightwave Technology*, vol. 29, no. 21, pp. 3340–3349, 2011.

[30] B. D. Manharbhai, A. K. Garg, and V. Janyani, "A flexible remote node architecture for energy efficient direct ONU internetworking in TDM PON," in *Proc. International Conference on Computer, Communications and Electronics (Comptelix)*, 2017.

[31] H. Lin, C. Lai, Y. Kao, and T. Wang, "ONU-based Decentralized Dynamic Bandwidth Allocation (OD-DBA) Scheme over WDM Long-Reach EPONs," in *Proc. International Conference on Information Networking (ICOIN)*, 2016.

[32] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy Consumption in Optical IP networks," *J. Lightwave Technology*, vol. 27, no. 13, pp. 2391–2403, 2009.

[33] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Energy Efficiency in Telecom Optical Networks," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 4, pp. 441–458, 2010.

[34] A. F. Pakpahan, I. Hwang, and A. Nikoukar, "OLT energy savings via software-defined dynamic resource provisioning in TDMA-PONs," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 11, pp. 1019–1029, 2017.

[35] A. F. Pakpahan, "Adaptive ONU energy-saving via software-defined mechanisms in TDMA-PON," in *Proc. 10th International Conference on Ubiquitous and Future Networks (ICUFN)*, 2018.

[36] S. Nishihara, M. Hajduczenia, H. Mukai, H. Elbakoury, R. Hirth, and M. Kimura, "Power-saving methods with guaranteed service interoperability in Ethernet passive optical networks," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 110–117, 2012.

[37] A. Otaka, "Power saving ad-hoc report," *Contribution to IEEE P802.3av Task Force*, 2008.

[38] J. Mandin, "EPON Power Saving via Sleep mode," *IEEE P802.3av 10GEPON Task Force Meeting*, 2008.

[39] S. W. Wong, L. Valcarenghi, S. H. Yen, D. R. Campelo, S. Yamashita, and L. Kazovsky, "Sleep mode for energy saving PONs: advantages and drawbacks," in *IEEE Globecom Workshops*, 2009.

[40] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[41] R. K. Naha and S. Garg, "Fog computing - survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, pp. 47980–48009, 2018.

[42] Guenter Klas, "Edge Computing and the Role of Cellular Networks," *Computer*, vol. 50, no. 10, pp. 40–49, 2017.

[43] S. Kitanov, E. Monteiro, and T. Janevski, "5G and the Fog - Survey of related technologies and research directions," in *Proc. of the 18th Mediterranean Electrotechnical Conference (MELECON)*, 2016.

[44] S. Kitanov and T. Janevski, "State of the Art: Fog Computing for 5G Networks," in *Proc. 24th Telecommunications Forum (TELFOR)*, pp. 1–4, 2016.

[45] A. Aljumah, “Fog computing and security issues - a review,” in *Proc. 7th International Conference on Computers Communications and Control (ICCCC)*, 2018.

[46] V. Backaitis, “Cloud Computing Takes a Back Seat to Edge Computing. Or Is It Fog?,” CMSWire. [Online]. Available: https://www.cmswire.com/internet-of-things/cloud-computing-takes-a-back-seat-to-edge-computing-or-is-it-fog/.

[47] I. Stojmenovic, “Fog computing: A cloud to the ground support for smart things and machine-to-machine networks,” in *Proc. Australasian Telecommunication Networks and Applications Conference (ATNAC)*, 2014.

[48] I. Al Ridhawi, Y. Kotb, and Y. Al Ridhawi, “Workflow-Net Based Service Composition Using Mobile Edge Nodes,” *IEEE Access*, vol. 5, pp. 23719–23735, 2017.

[49] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha, “Service composition for mobile environments,” *Mobile Networks and Applications*, vol. 10, pp. 435–451, 2005.

[50] S. Deng, L. Huang, H. Wu, and Z. Wu, “Constraints-Driven Service Composition in Mobile Cloud Computing,” in *Proc. IEEE International Conference on Web Services (ICWS)*, 2016.

[51] N. C. H. Ngoc, D. Lin, T. Nakaguchi, and T. Ishida, “QoS-aware service composition in mobile environments,” in *Proc. IEEE 7th International Conference on Service-Oriented Computing and Applications (SOCA)*, 2014.

[52] D. Kasamatsu, M. Kumar, and P. Hu, “Service Compositions in Challenged Mobile Environments under Spatiotemporal Constraints,” in *Proc. IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017.

[53] S. Deng, L. Huang, J. Taheri, J. Yin, M. C. Zhou, and A. Y. Zomaya, “Mobility-aware service composition in mobile communities,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 3, pp. 555–568, 2017.

[54] S. Deng, H. Wu, W. Tan, Z. Xiang, and Z. Wu, "Mobile Service Selection for Composition : An Energy Consumption Perspective," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 3, pp. 1–13, 2017.

[55] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[56] K. Intharawijitr, K. Iida, and H. Koga, "Analysis of fog model considering computing and communication latency in 5G cellular networks," in *Proc. IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom)*, 2016.

[57] J. Oueis, E. C. Strinati, and S. Barbarossa, "The Fog Balancing: Load Distribution for Small Cell Cloud Computing," in *Proc. 2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–6, 2015.

[58] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, "Small Cell Clustering for Efficient Distributed Fog Computing : A Multi-user Case," in *Proc. 2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2015.

[59] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[60] R. Mahmud and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," *Distributed, Parallel, and Cluster Computing*, pp. 1–28, 2016.

[61] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *J. Network and Computer Applications*, vol. 40, no. 1, pp. 325–344, 2014.

[62] A. Reaz, V. Ramamurthi, and M. Tornatore, "Cloud-over-WOBAN (CoW): An offloading-enabled access network design," in *Proc. IEEE International Conference on Communications (ICC)*, 2011.

[63] W. Zhang, B. Lin, Q. Yin, and T. Zhao, "Infrastructure deployment and optimization of fog network based on MicroDC and LRPON integration," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 579–591, 2017.

[64] B. P. Rimal, D. Pham Van, and M. Maier, "Cloudlet Enhanced Fiber-Wireless Access Networks for Mobile-Edge Computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3601–3618, 2017.

[65] Y. Luo, F. Effenberger, and M. Sui, "Cloud computing provisioning over Passive Optical Networks," in *Proc. 1st IEEE International Conference on Communications in China (ICCC)*, 2012.

[66] B. P. Rimal, D. Pham Van, and M. Maier, "Mobile-edge computing vs. centralized cloud computing in fiber-wireless access networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2016.

[67] S. Mondal, G. Das, and E. Wong, "CCOMPASSION: A Hybrid Cloudlet Placement Framework over Passive Optical Access Networks," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2018.

[68] E. Wong, M. Mueller, M. P. I. Dias, C. A. Chan, and M. C. Amann, "Energy-efficiency of optical network units with vertical-cavity surface-emitting lasers," *Optical Express*, vol. 20, no. 14, pp. 14960–14970, 2012.

[69] E. Wong, M. Mueller, and M. C. Amann, "Characterization of energy-efficient and colorless ONUs for future TWDM-PONs," *Optical Express*, vol. 21, no. 18, pp. 20747–20761, 2013.

[70] B. Schrenk, S. Member, F. B. Bo, J. Bauwelinck, J. Prat, and J. A. Lazaro, "Energy-Efficient Optical Access Networks Supported by a Noise-Powered Extender Box," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 17, no. 2, pp. 480–488, 2011.

[71] L. Valcarenghi et al., “Energy efficiency in passive optical networks: where, when, and how,” *IEEE Network*, vol. 26, no. 6, pp. 61–68, 2012.

[72] L. Valcarenghi, M. Presi, G. Contestabile, P. Castoldi, and E. Ciaramella, “Impact of Modulation Formats on ONU Energy Saving,” in *Proc. 36th European Conference and Exhibition on Optical Communication (ECOC)*, 2010.

[73] R. Kubo et al., “Study and Demonstration of Sleep and Adaptive Link Rate Control,” *IEEE/OSA J. Optical Communications and Networking*, vol. 2, no. 9, pp. 716–729, 2010.

[74] L. Shi, S.-S. Lee, and B. Mukherjee, “An SLA-based energy-efficient scheduling scheme for EPON with sleep-mode ONU,” in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference*, 2011.

[75] S. Wong, S. Yen, P. Afshar, S. Yamashita, and L. G. Kazovsky, “Demonstration of Energy Conserving TDM-PON with Sleep Mode ONU using Fast Clock Recovery Circuit,” in *Proc. Conference on Optical Fiber Communication collocated National Fiber Optic Engineers Conference (OFC/NFOEC)*, 2010.

[76] M. Fiammengo, A. Lindström, P. Monti, L. Wosinska, and B. Skubic, “Experimental Evaluation of Cyclic Sleep with Adaptable Sleep Period Length for PON,” in *Proc 37th European Conference and Exhibition on Optical Communications (ECOC)*, 2011.

[77] D. P. Van et al., “Energy-saving framework for passive optical networks with ONU sleep/doze mode,” *Optical Express*, vol. 23, no. 3, pp. 1–14, 2015.

[78] Y. Yan et al., “Energy Management Mechanism for Ethernet Passive Optical Networks (EPONs),” in *Proc. IEEE International Conference on Communications (ICC)*, 2010.

[79] D. P. Van, B. P. Rimal, and M. Maier, “Fiber optic vs. wireless sensors in energy-efficient integrated FiWi smart grid networks: An energy-delay and TCO comparison,” in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, 2016.

[80] A. R. Dhaini, P. H. Ho, and G. Shen, "Toward green next-generation passive optical networks," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 94–101, 2011.

[81] D. P. Van, B. P. Rimal, M. Maier, and L. Valcarenghi, "ECO-FiWi: An Energy Conservation Scheme for Integrated Fiber-Wireless Access Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3979–3994, 2016.

[82] A. Helmy, H. Fathallah, and H. Mouftah, "Interleaved Polling versus Multi-Thread Polling for Bandwidth Allocation in Long-Reach PONs," *IEEE/OSA J. Optical Communications and Networking*, vol. 4, no. 3, pp. 210–218, 2012.

[83] B. P. Rimal, M. Maier, and M. Satyanarayanan, "Experimental Testbed for Edge Computing in Fiber-Wireless Broadband Access Networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 160–167, 2018.

[84] M. Jia, J. Cao, W. Liang, and S. Member, "Optimal Cloudlet Placement and User to Cloudlet Allocation in Wireless Metropolitan Area Networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2017.

[85] A. From and O. Call, "Green Cloudlet Network: A Distributed Green Mobile Cloud Network," *IEEE Network*, vol. 31, no. 1, pp. 64–70, 2017.

[86] M. Hossen and S. Saha, "Thread guaranteed algorithm for real time traffic in multi-Threaded polling of PON-based open access network," in *Proc. International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017.

[87] H. Wang et al., "An inter multi-thread polling for bandwidth allocation in Long-Reach PON," in *14th International Conference on Optical Communications and Networks (ICOCN)*, 2015.

[88] Y. Yan and L. Dittmann, "Analysis of sleep-mode downlink scheduling operations in EPON systems," in *8th International Conference on Information, Communications and Signal Processing (ICICS),* 2011.

[89] M. P. McGarry, M. Reisslein, and M. Maier, "Ethernet Passive Optical Network Architectures and Dynamic Bandwidth Allocation Algorithms," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 46–60, 2008.

[90] S. Mondal, G. Das, and E. Wong, "Efficient cost-optimization frameworks for hybrid cloudlet placement over fiber-wireless networks," *IEEE/OSA J. Optical Communications and Networking*, vol. 11, no. 8, 2019.

[91] Spirent Communications, "Test methodology journal: IMIX (Internet Mix)," *White Paper*, Mar. 2006.